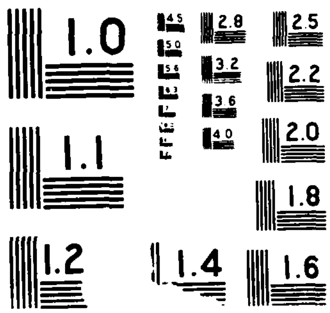
AD-A194 516 ARCHITECTURE AND INTEGRATION REQUIREMENTS FOR AN ULCE 1/2
(UNIFIED LIFE CYCLE (U) INSTITUTE FOR DEFENSE ANALYSES
ALEXANDRIA VA M L BREI ET AL APR 88 IDA-P-2063
UNCLASSIFIED IDA/HQ-87-32844 MDA903-84-C-0031 F/G 5/1 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.2
1.4
1.6

Copy 13 of 189 copies

DTIC FILE COPY

(2)

IDA PAPER P-2063

ARCHITECTURE AND INTEGRATION REQUIREMENTS FOR AN ULCE DESIGN ENVIRONMENT

DTIC ELECTE JUN 1 7 1988 S D
D

ML Brei
William E. Cralley
David Dierolf
David J. Owen
Karen J. Richter
Ed Rogan, Lockheed Company

AD-A194 516

April 1988

Prepared for
Office of the Under Secretary of Defense for Acquisition

Supported by
Air Force Human Resources Laboratory
Wright-Patterson AFB, Ohio

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

IDA
INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311

IDA Log No. HQ 87-32844

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A194516

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY DD Form 254 dated 1 October 1983	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) IDA Paper P-2063	5. MONITORING ORGANIZATION REPORT NUMBER (S)
6a. NAME OF PERFORMING ORGANIZATION Institute for Defense Analyses 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION OSD, OUSDRE, DoD-IDA Management Office
6c. ADDRESS (CITY, STATE, AND ZIP CODE) 1801 North Beauregard Street Alexandria, Virginia 22311	7b. ADDRESS (CITY, STATE, AND ZIP CODE) 1801 North Beauregard Street Alexandria, Virginia 22311
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of the Under Secretary of Defense (R&AT)/ET 8b. OFFICE SYMBOL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA 903 84 C 0031
8c. ADDRESS (City, State, and Zip Code) The Pentagon Washington, DC 20301-3080	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT: — PROJECT NO.: — TASK NO.: T-D6-508 ACCESSION NO. WORK UNIT: —

11. TITLE (Include Security Classification)

ARCHITECTURE AND INTEGRATION REQUIREMENTS FOR AN ULCE DESIGN ENVIRONMENT

12. PERSONAL AUTHOR(S)

ML Brei, William E. Cralley, David Dierolf, David J. Owen, Karen J. Richter, Ed Rogan

13. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 3/87 TO 11/87	April 1988	171

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Unified Life Cycle Engineering, ULCE, Engineering Design Process, Supportability, Producibility, Trade-Offs, Meta-Design, Quality, Life Cycle Cost, Computer-Aided Engineering, CAE, Object-Oriented, Data Base

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The goal of Unified Life Cycle Engineering is to develop an advanced design environment that allows considerations of producibility and supportability to be integrated into the design process in a timely fashion, i.e., early in the design process, along with the usual considerations of performance, cost, and schedule. This paper reports the results of a study to develop the needed architecture for a design process that would implement Unified Life Cycle Engineering (ULCE) and analyze the problems of assimilation, interpretation, and integration of diverse data bases and analytical tools. The developed architecture addresses multiple levels of subsystem hierarchy and incorporates concurrency in the consideration of factors related to producibility, supportability, performance, cost, and schedule in the design process. Requirements and specifications for an executive and a control system to implement an ULCE architecture are developed and trends in research relevant to integration problems that will impact ULCE systems development are identified.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS REPORT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22C. OFFICE SYMBOL

DD FORM 1473, 84 MAR

C.S. 4-22-88

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

IDA PAPER P-2063

ARCHITECTURE AND INTEGRATION REQUIREMENTS FOR AN ULCE DESIGN ENVIRONMENT

ML Brei
William E. Cralley
David Dierolf
David J. Owen
Karen J. Richter
Ed Rogan, Lockheed Company

April 1988

IDA

INSTITUTE FOR DEFENSE ANALYSIS

Contract MDA 903 84 C 0031
Task T-D6-508

PREFACE

This report was prepared by the Institute for Defense Analyses (IDA) for the Office of the Under Secretary of Defense for Acquisition and the Air Force Human Resources Laboratory, under contract number MDA 903 84 C 031, Task Order T-D6-508, Architecture and Integration Requirements - ULCE.

The issuance of this report meets the specific tasks of "[examining] alternative architectures for the design process needed to implement ULCE and [analyzing] the problems of assimilation, interpretation, and integration of diverse data bases and analytical tools which must be solved to implement such architectures."

ACKNOWLEDGMENTS

The following people at Lockheed Aeronautical Systems Company contributed to this report:

John Brooks	Tom McBrayer
Mitchell Cline	Ed Lowery
Paul Cole	Henry Smith
Ken Johnson	Terry Stancil

CONTENTS

PREFACE iii

ACKNOWLEDGMENTS v

GLOSSARY xiii

EXECUTIVE SUMMARY ES-1

A. Introduction ES-1

1. Background ES-1
2. Task Objectives ES-2
3. Approach ES-2
4. Identification of Target System for Study ES-3

B. Current Design Process ES-3

1. Limitations of the Current Process ES-4
2. Design Goals, Criteria, Requirements, and Tradeoffs ES-5

C. ULCE Design Process Architecture ES-6

1. Top Level Overview ES-6

D. Software Requirements ES-11

E. Conclusions and Recommendations ES-15

1. General Philosophy of the ULCE Design Process ES-15
2. Feasibility of Implementation ES-16
3. Recommendations ES-17

F. Next Steps ES-19

A. INTRODUCTION 1

1. Background 1
2. Task Objective 2

3. Some Aspects of an ULCE System 2
4. Scope of ULCE 3
5. ULCE System Considerations 3
6. Task Approach and Report Organization 7

B. TARGET SYSTEM DESCRIPTION 9

C. CURRENT DESIGN PROCESS 13

1. Overview 13
2. Procedure Flow 16
3. Information Flow 25
4. Organization 25
5. Conclusions 29

D. DESIGN GOALS, CRITERIA, REQUIREMENTS, AND TRADEOFFS 31

1. Requirements 31
2. Detailed Design Considerations 35
3. Tradeoffs 38
4. Producibility 38
5. Supportability 46
6. Analytical Tools 47
7. Decision Process for Hierarchical Assignment of Goals and Requirements 50
8. Arbitration Among Conflicting Design Goals 51
9. Conclusions and Observations 53

E. ULCE ARCHITECTURE 55

1. Introduction 55
2. Top Level ULCE Architecture 59
3. ULCE Data Flow 61
4. One-Level-Down ULCE Architecture 65
5. Technology Approach to the Proposed ULCE Architecture 80
6. Application of the ULCE Architecture 83

F. SOFTWARE REQUIREMENTS FOR THE ULCE ARCHITECTURE 87

1. Introduction 87
2. Current Computer-Aided Engineering Software 88

3. Implementing Instantiations of the Design Process ... 91
4. Implementing the Design-in-Progress ... 99
5. Implementing the Meta-Design Process ... 103
6. Summary ... 104

G. AN EVALUATION OF THE PROPOSED ULCE ARCHITECTURE ... 109
1. Introduction ... 109
2. General Philosophy of the ULCE Design Process ... 109
3. Technical Evaluation ... 111
4. Likely Improvements in Design Quality ... 116
5. Potential Life Cycle Cost Savings ... 118

H. CONCLUSIONS AND RECOMMENDATIONS ... 121
1. Conclusions ... 121
2. Recommendations ... 124

REFERENCES ... 127

DISTRIBUTION ... DL-1

LIST OF FIGURES

ES-1. ULCE Data Flow ES-7
ES-2. Make Design Decisions Procedure Flow ES-12
A-1. Military Hardware Life Cycle 4
A-2. Functional Hierarchy 5
A-3. Typical Reliability/Maintainability Analyses-Design Function 6
B-1. High Sink Rate Gear 10
B-2. High Sink Rate Landing Gear on the C-130 HTTB 11
C-1. Iteration in Current Design Process 13
C-2. Iterative Redesign Loops in Design Decision Sequence 14
C-3. Details of Iterative Loops 15
C-4. Landing Gear Preliminary Design 17
C-5. Landing Gear Post-Contract Design 18
C-6. Producibility in the Design Process 23
C-7. Simplified Supportability Design Process 24
C-8. Preliminary Design Phase Data Flow 26
C-9. Post Contractual Design Data Flow 27
C-10. Organizational Structure 28
D-1. Supportability Analysis Tools 49
D-2. Comparative Evaluations 50
D-3. Hierarchy of Tradeoff Elements 51
D-4. Trade Study Process 52
E-1. "Freeze and Squeeze" in the Current Sequential Design Process 56
E-2. Supportability Technical Tasks 57
E-3. An Example of Information Loss 58
E-4. ULCE Procedure Flow 59
E-5. ULCE Data Flow 61
E-6. Design Concepts 64

E-7. Generate Design Alternatives Procedure Flow 65
E-8. Generate Design Alterantives Data Flow 66
E-9. Functional Analysis 67
E-10. Comparison of Strokes of Existing and High Sink Rate Struts 68
E-11. Gear Bottoming Prevented 69
E-12. MLG Minor Modified and Hub Assembly 69
E-13. System Definition 70
E-14. Plan Design Decision Process Procedure Flow 71
E-15. Plan Design Decision Process Data Flow 72
E-16. Landing Gear Design 73
E-17. Decision Task Network 74
E-18. Make Design Decisions Procedure Flow 76
E-19. Alternatives 78
E-20. Make Design Decisions Data Flow 79
E-21. ULCE Design Process Sequence 85

GLOSSARY

AC	Aircraft
AFLC	Air Force Logistic Command
ANSI	American National Standards Institute
AO	Operational availability
CAD	Computer-aided design
CAE	Computer-aided engineering
CALS	Computer-aided acquisition and logistics support
CDR	Critical design review
CG	Center of gravity
CME	Chief manufacturing engineer
DBMS	Data base management system
DoD	Department of Defense
DOL	Dispersed operation locations
EDIF	Electronic Design Interchange Format
EIA	Electronic Industries Association
EIS	Engineering Information System
FMECA	Failure mode, effect, and criticality analysis
HSR	High sink rate
HTTB	High technology testbed
IEEE	Institute for Electrical and Electronic Engineers
IGES	Initial Graphics Exchange Specification
ilities	those attributes of a product that in totality are measures of effectiveness or of fitness of use for accomplishing desired objectives
LSA	Logistics support analysis
LRU	Line replaceable unit
M	Maintainability
MAP/TOP	Manufacturing Automation Protocol/Technical Office Protocol
MCR	Mission capable rate
MLG	Main landing gear
MIL-STD	Military standard
MOLD	Multi-level optimization using linear decomposition
MTBCE	Meantime between critical events
MTBF	Meantime between failures
MTTR	Meantime to repair

NASA	National Aeronautics and Space Administration
NBC	Nuclear, biological, and chemical
NLG	Nose landing gear
NRLA	Network repair level analysis
PD	Preliminary design
PDES	Product Data Exchange Specification
PDR	Preliminary design review
R	Reliability
RAMCAD	Reliability, Availability and Maintainability Computer-Aided Design
RFP	Request for proposal
S	Supportability
SDTR	Supportability design to requirements
SGR	Sortie generation rate
SOW	Statement of work
SQL	Standard query language
STAIRS	Storage and Information Retrieval System
TBO	Time between overhaul
ULCE	Unified Life Cycle Engineering
VHDL	VHSIC Hardware Design Language

EXECUTIVE SUMMARY

A. INTRODUCTION

1. Background

Recently, the issue of quality in DoD weapon systems has been receiving increased attention. While most of this attention has been focused on manufacturing issues, there is a growing realization that while quality can be lost in the manufacturing phase, it cannot be gained there. The inherent quality of a weapon system, or of any other product for that matter, is determined by its design. In turn, the design of the product determines the extent of supportability of the product in the field by how well it satisfies the requirements of the people who use the product and of the environment in which the product is used.

These considerations have been recognized by OSD and the Services, and programs are now underway which will address the need for better design quality in DoD systems. One such program is the Air Force Systems Command Unified Life Cycle Engineering (ULCE) Program, a Project Forecast-II research and development initiative, whose stated goal is:

> "...to develop, demonstrate, and transfer to application the techniques and technologies needed to provide advantageous computerized integration of the procedures dealing with designing for producibility and supportability with those dealing with designing for performance, cost, and scheduling..." [Ref. 1]

The ULCE program contains a number of individual R&D projects, two of which (Decision Support Requirements for an ULCE Design Environment and Architecture and Integration Requirements for ULCE) have been conducted by IDA. This report contains the results of the latter project, which was conducted as task MDA 903 84C 0031, T-D6-508, sponsored by the Air Force Human Resources Laboratory and the office of the Undersecretary of Defense for Acquisition.

2. Task Objectives

The overall objective was to develop the needed architecture for a design process which would implement ULCE and analyze the problems of assimilation, interpretation, and integration of diverse data bases and analytical tools. This effort included:

(1) Investigating architectures for the design process which would address multiple levels of subsystem hierarchy and incorporate concurrency in the consideration of factors related to producibility, supportability, performance, cost, and schedule in the design process.

(2) Developing requirements and specifications for an executive and a control system to implement an ULCE architecture.

(3) Identifying trends in research relevant to integration problems which will impact ULCE systems development.

3. Approach

To ensure that the architecture for ULCE developed in this study would reflect real world considerations and would be responsive to the many problems which must be addressed by the design community when designing a complicated system, it was decided that the architecture would be developed in the context of a specific hardware design problem and that IDA would seek assistance from a major defense contractor in developing the architecture. During the early phase of the study, IDA personnel visited a number of defense contractors and universities in order to get a good picture of the state-of-the-art in integration of engineering functions and advanced design technologies. Based on these visits and on initial study planning, a request for proposal (RFP) for technical support to IDA was developed and distributed to a number of contractors. Lockheed Aeronautical Systems Company (Georgia) was selected to participate in the study based on the responses to this RFP; the two aspects of ongoing Lockheed activities that were felt to be particularly germane to the ULCE objectives are listed below.

- They have instituted a long-term, planned approach to integration of computer-aided tools for all engineering activities.

- They have recognized the need to provide concrete measures of supportability and are developing Supportability Design-To Requirements (SDTR's) to be input in the early stages of the design process.

4. Identification of Target System for Study

Several hardware and hardware/software systems were proposed as candidate target systems for evaluation of the current design process and for use as a testbed for development of the ULCE architecture. The High Sink Rate (HSR) Landing Gear for the C-130 aircraft was chosen as the target system based on the following considerations:

- The HSR landing gear design reflects the effects of changing operational and support requirements. AirLand Battle 2000 and the R&M 2000 initiatives lead to requirements to operate from dispersed locations having rough, short fields. The C-130 will be required to land at a 15 feet/sec sink rate at 130,000 lbs, versus the current requirement of a sink rate of 9 feet/sec. Moreover, the new gear must require minimum support equipment, facilities, and skilled maintenance personnel and must be maintainable in a nuclear, biological, and chemical (NBC) environment.
- While most of the attention in the ULCE program has been directed towards electronic design problems, more attention needs to be given to the mechanical and structural areas. The landing gear is a good candidate for such attention.
- The landing gear design process highlights the interactions between prime and subcontractors. The HSR gear can be broken down into five levels of design hierarchy. Communication between engineers and subcontractors working at each of these levels is critical to a successful ULCE design process for landing gear. Moreover, requirements for the HSR gear impact significantly on overall aircraft system design. The high sink rate requirement will necessitate aircraft structural changes to absorb increased loads upon landing.

B. CURRENT DESIGN PROCESS

Before embarking on the development of an ULCE architecture for landing gear design, the study team examined the current way landing gears are designed at Lockheed, with a view toward determining problems with the current process which should be alleviated under the ULCE architecture. The team also examined the nature of the design requirements, goals, and tradeoffs which characterize landing gear design in order to determine specific features in the ULCE architecture needed to handle these considerations. The primary sources for information for this investigation were discussions with design engineers at Lockheed and Lockheed's standard landing gear design handbook [Ref. 4.]

1. Limitations of the Current Process

The following points characterize the current landing gear design process in terms of which factors drive it and what limitations in process and product result.

- *The process is driven by scheduled deliverable data items.* Pressure to turn out drawings and specifications leads to a design process which is optimized for a depth-first search of the design space leading to a rapid build up of design definition and detail.

- *Definition of design detail is engineering labor-intensive (and therefore costly.)* Much of the required detail is generated manually. However, even with computer aided design and drafting tools, such as CADAM^{tm}, considerable engineering time is required to produce drawings and other documentation.

- *As a result of the above considerations, the design process is characterized by a relatively rigid sequence of design decisions.* This sequence currently is not based on a prioritization of requirements, but on minimizing the cost of the process (by minimizing iterative redesign loops and moving to the end of the design process those decisions requiring considerable design definition and detail).

- *Producibility and Supportability considerations enter relatively late in the design process.* This is primarily because such considerations require a level of design definition which is not available early in the current process.

- *Production planning, logistics support analysis, development of maintenance concepts, and development of reliability and maintainability allocations and goals occur outside of the current design process.* These factors impact the current design process through the requirements definition and design review procedures. While requirements definition may require development and critique of design concepts by engineering specialists, the resulting requirements are usually communicated to designers without the additional context of these design concepts. The designers are left to their own devices to determine why a particular *ility*-related requirement has been placed on the design. Direct communication and participation by specialty engineers in the design concept development and definition activities is relatively limited.

- *Design data is fragmented.* The total design is described by a number of separate pieces of information such as sketches, stick diagrams, various specifications, computer files containing 3D solid models, 2D drawings and views, finite element models and parts lists, and formal hard copy drawings which are supplied to manufacturing or to subcontractors for fabrication. Using current procedures, it is all but impossible to maintain consistency among all of the different representations of various design aspects. Inconsistencies lead to design errors and costly redesign activities.

- *Information is lost as the design progresses.* While implementation concepts are formally documented, descriptions of intended functions (and unintended functions, if they have been identified) may be only informally documented, or not at all. Therefore, design intent is not always adequately conveyed as the design moves downstream. The reason certain features are present in the design must be conveyed along with the geometry, material specifications, tolerances, etc. Otherwise, changes may be made to the design to improve certain downstream functions (such as manufacturability or supportability) which compromise the performance characteristics of the design.

2. Design Goals, Criteria, Requirements, and Tradeoffs

A multitude of design requirements, goals, and criteria face the landing gear design team. While some requirements come directly from the RFP or other contractual documents, many are internally imposed guidelines based on design experience. Also, a number of Mil-Specs and standards may be required, each referring to other standards and specifications and greatly multiplying the number of considerations which must be taken into account in designing the landing gear. When considerations of producibility and supportability are brought in, a veritable explosion of requirements ensues. Management and tracking of such large numbers of requirements will be a major problem under ULCE.

In the research leading to this study, the study team identified the following numbers of requirements (which is surely only a subset of the total number):

General performance requirements and criteria:

- 46 general requirements and guidelines internally imposed by Lockheed.
- 28 Mil-Specs and Standards which are applicable to landing gear design--each containing dozens to hundreds of individual requirements.
- 26 checklist items in detailed design internally imposed at Lockheed.
- 8 common trade studies performed in landing gear design

Producibility related requirements and criteria include:

- 13 general producibility considerations for forged parts
- 11 criteria for eliminating or combining parts.
- 19 requirements relating to material processing
- 31 requirements relating to machining
- 31 requirements relating to forging
- 5 criteria related to part identification

- 4 criteria related to standardization
- 13 general requirements related to installation of assemblies
- 5 requirements relating to eliminating or combining parts
- 4 criteria relating to assembly
- 9 criteria relating to sub-assemblies
- 24 criteria related to fasteners
- 7 criteria related to standards

Supportability requirements:

- A sample of 13 supportability design-to requirements was identified

The requirements identified above are only a sampling of the total requirements facing the designer. While a number of these requirements are quantitative performance requirements, many are qualitative in nature. Qualitative requirements and criteria dominate in considerations of producibility and supportability. A decision process which effectively handles both qualitative and quantitative requirements is essential for ULCE design process.

C. ULCE DESIGN PROCESS ARCHITECTURE

As noted above, the current design process suffers several limitations--lack of direct participation of engineering specialists in the development of design concepts, a relatively rigidly structured design-decision process which is driven by cost and schedule rather than requirements, fragmented design information, and loss of design information as the process transitions from one phase to the next. Each of these problems are addressed in the ULCE architecture.

1. Top Level Overview

Figure ES-1 illustrates the ULCE architecture from a top level perspective. The ovals in this figure represent procedures which are carried out by the design team (with computer support), while the rectangles represent information bases which support the design process. The arrows represent interactions between the design team and data in the information bases. Each component of the architecture will be described below.

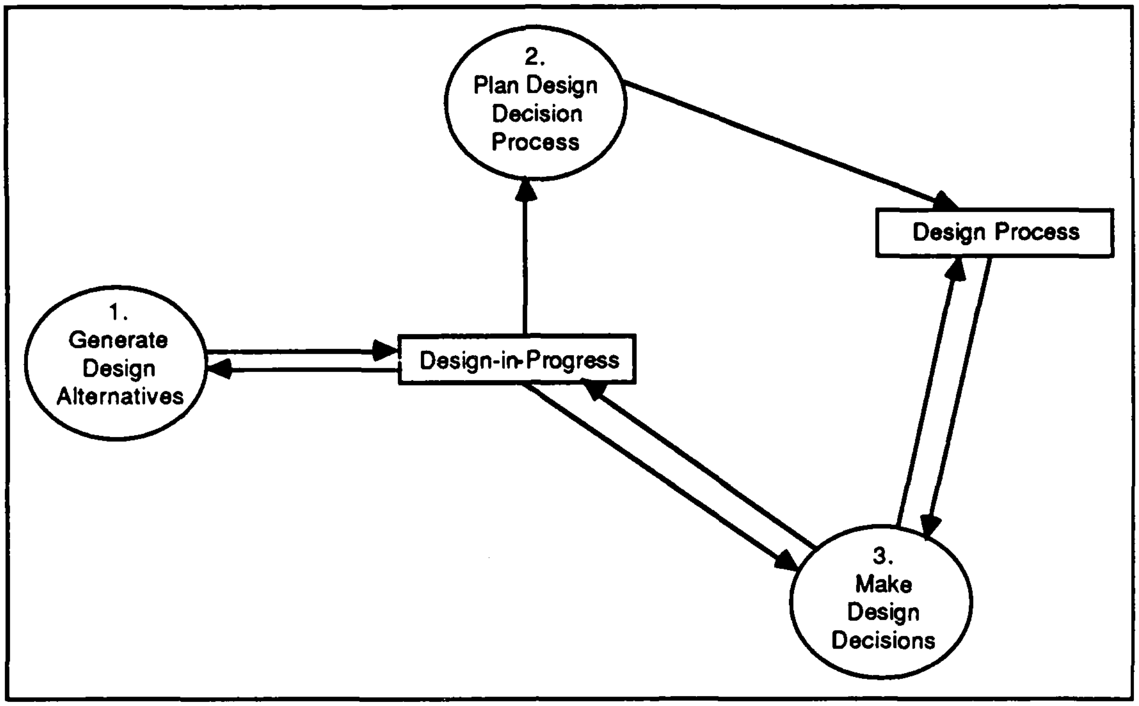


Figure ES-1. ULCE Data Flow

a. Generate Design Alternatives

A key element of the ULCE architecture is the mechanism for incorporating specialty considerations (including producibility and supportability) early in the design process. In order to accommodate this, the ULCE architecture must address how design alternatives can be generated by engineering specialists who are currently not considered to be designers. In the early stages of design, where concepts are being generated, it is critical that specialty engineers be a part of the design team and that they be encouraged to contribute design concepts which are compatible with production and support requirements. While such concepts may not satisfy all design requirements related to performance characteristics, by allowing them to be "put on the table" the chances of incorporating some of their features into the final design is increased and the concept will be better developed from the standpoint of these later considerations than one produced by a team without members who are experienced in these specialty areas.

The proposed architecture provides an integrated process for creative generation of design alternatives by all members of the design team. This process will bring together tools currently used by preliminary design engineers, landing gear designers, producibility

and supportability specialists, and systems engineers. All members of the design team will be allowed to suggest design ideas. Design alternatives will include maintenance and production concepts as well as ideas for performance related system elements that are currently considered part of the design definition. These design alternatives should be responsive to requirements, but need not simultaneously meet all requirements.

b. Plan Design Decision Process

One proposal that has been advanced for achieving better designs that are more easily produced and supported is to require producibility and supportability analyses to be performed earlier in the design process. This approach is not likely to solve the whole problem, however, because designs are properly balanced by decision making, not analysis. The timetable for making these decisions will be program-specific, depending on requirements flow, technology, decision support, and analysis tools. ULCE must address the need for a *Meta Design Process* to design the design decision-making process. Emphasis on the design-decision timetable and recognition of the role of engineering analysis in evaluating design alternatives will help to avoid problems in the current design practice specifications.

The design decision process plan must consider the implications of all the relationships among attributes of the design alternatives. For relatively simple designs, this can be done by a small group of design integrators. Although the process of design decision planning usually is not considered explicitly for relatively small scale efforts, the ability to understand the implications of a complex set of interacting requirements and design attributes is a mark of design genius. The effectiveness of an individual design genius at integrating the efforts of a large group of engineers on a complex, multidisciplinary problem is limited, however, by program schedule constraints. Thus, the ULCE architecture explicitly provides for computer support for the process of identifying and scheduling design decisions affecting integration of the design.

Software support will be used to decompose a symbolic representation of the alternatives for design concept attributes into discrete design decision tasks and the interfaces between those tasks. A decision timetable can then be developed by considering the amount of time needed to apply simulation and analysis tools to evaluate the alternatives. This *Meta Design*, the Plan Design Decision Process step, is in fact a procedure for defining and building the next step of the architecture and plays a principal

role in facilitating the design optimization necessary to obtain properly balanced designs within the ULCE architecture.

c. Make Design Decisions

The design decision process plan identifies design decisions involving a mixture of qualitative and quantitative considerations. The process of making these decisions will be done interactively, using an integrated tool kit of knowledge-based systems, design optimization techniques, and methods of measuring design attributes and ranking designs based on these measurements. The computing environment supporting this step must allow for capture of substantiation developed by the design team to support design decisions. This substantiating information will be used for preliminary and critical design reviews and to support iteration with other decisions when required.

The design decision making tasks will specify a preferred alternative for each of several design attributes. If alternatives can be specified that fully meet requirements, the design team will move on to other design decisions. However, if no suitable alternative is found, the impact of prior decisions on the current decision is assessed, and these prior decisions would be iterated based on this assessment.

d. Design-In-Progress Information Base

The Design-in-Progress is the core of the proposed ULCE architecture. The Design-in-Progress is the repository of all knowledge relating to the product being designed that has been generated in the course of design activity. As the design proceeds through its various stages, the amount of information or knowledge about the design continues to grow. This information includes not only an increasing level of detailed information on product description, it also includes historical data on analyses and design decisions. The Design-in-Progress will provide ready access to all of the current information pertaining to a particular design which may be needed by any of the member of the design team.

Object-oriented technology will play a major role in development of the Design-in-Progress. Interactions of the design team and the Design-in-Progress will include: (1) defining a design object, (2) instantiating a design object, and (3) invoking a method (a computer program or procedure) associated with a design object or an instance of an object. In the Generate Design Alternatives phase of the architecture, objects corresponding to requirements, functions, and implementations of alternative design concepts will be

defined. These objects will then be accessed by the other computer programs which are used to provide support for the design process planning phase of the architecture.

The methods associated with design objects include such tools as procedures for simulating or analyzing aspects of the design and means of constructing a representation (a view of the object). Once a design concept has been defined, the representation in any one of the views can be accomplished by executing the method associated with generating that view. This will provide the design team with a powerful parametric design capability.

e. Design Process Representation

The last component of the ULCE architecture consists of an object oriented representation of the set of design tasks which must be accomplished to carry out the design decision process and to arrive at an acceptable design. This information base is constructed by software which extracts design attributes and engineering relationships from the design objects which are contained in the Design-in-Progress. These relationships are used to decompose the overall task into smaller decision tasks. These tasks are then instantiated by the design decision process planning software, and these instances collectively make up the design decision process computing environment. Each instance of a design task provides access to local decision support tools for use by the design team as part of the Make Design Decisions phase of the architecture and supports tracking of the global decision-making process (which is also a part of the decision-making procedure). The development of the design process information base and the computing environment will make the total integration of design activities and project management activities, which are critical to the development of properly balanced designs, possible .

f. Application of the ULCE Architecture

Design of large systems has by nature a sequential and iterative character which will not be changed by ULCE. The ULCE architecture, however, will provide changes in the way design is done, and this will result in significant benefits. First, the Design-in-Progress concept will greatly reduce the design cycle time by reducing design errors caused by communication failure and information loss. Second, the explicit consideration of design decision making brought about by the design decision process planning phase of the architecture (Meta Design) will highlight questions of producibility and supportability with sufficient timeliness to allow cost effective actions to be taken. Of course, without the integrated Design-in-Progress information base which gives near-real-time access to the

design to all members of the design team, including specialty engineers, the capability of the design team to generate good design alternatives is limited.

In Figure ES-2, the application of the ULCE architecture is shown in the context of a large system design. The design is shown pictorially as proceeding through the design process in discrete phases, with the Design-in-Progress being passed intact from each phase to the next. This sequence might be better described as proceeding down the design hierarchy with increasing detail being evaluated and added to the Design-in-Progress at each succeeding level. A key feature of the proposed ULCE architecture is that the overall process (Generate Alternatives, Plan the Design Decision Process, Make Design Decisions) is continually repeated unchanged as long as design activities continue.

An ancillary, but important, feature of the proposed architecture is that it has the flexibility to handle changes to the requirements or to the design at any stage of the design process. A change in requirements that might impact the conceptual design can be evaluated even when detailed design is underway. The process will allow the designers to quickly identify the functional and physical design features which will be affected by the change down through the hierarchy from system level to component level. The design process planning capability will allow a new process plan to be generated that takes into account which elements of the concept are different and which elements make maximal use of work that has been already accomplished. A new plan may be developed which will yield good estimates of the cost and schedule impact of changes in requirements at any point in the design process.

The Design-in-Progress will exist throughout the life cycle of a product. It will retain all of the information and knowledge about the product, including design decisions at the very first stages of concept development. Redesign for later modifications or product improvements, even 20 years after development, should pose no more difficulty than would be encountered in the original design activity.

D. SOFTWARE REQUIREMENTS

The ULCE design environment will be an integrated and layered set of automated design software which will support all aspects of the ULCE architecture. This software will facilitate the management of complex design parameters and relationships, the planning of the design process, and the execution of the design process. The following components must be developed to achieve this environment.

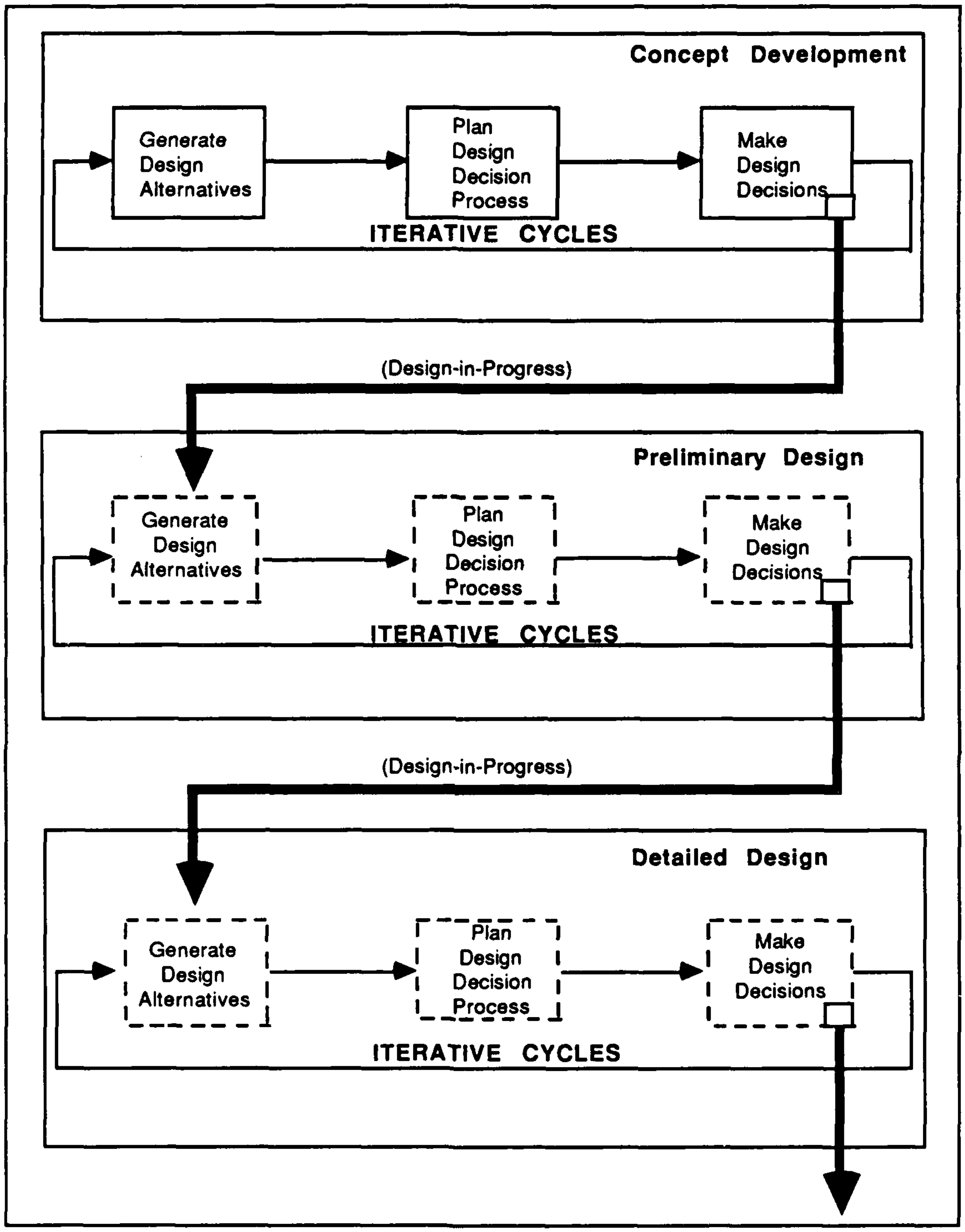


Figure ES-2. Make Design Decisions Procedure Flow

- *An advanced operating environment* . The operating environment provides intelligent execution of design tools, coordination of design management tasks, integration mechanisms, controlled access to all components of the ULCE environment, and mechanisms to shield users from underlying operating system details. To perform these functions, the operating environment makes use of many types of knowledge--knowledge of the product under design (generic), the design process (as planned), design management functions, the state and contents of the Design-in-Progress representation of the design, the information requirements for each of the automated tools, system utilities, and system configuration information. By using this knowledge, the operating environment will facilitate constant movement among the design process phases and levels of abstraction. Issues which must be addressed include:

  (1) How will the operating environment use the sets of complex knowledge to manage the environment?

  (2) What design management tasks are necessary?

  (3) How can external tools be integrated into the system without requiring massive software development efforts or limiting the capabilities of the tool?

  (4) How can the complexity of the underlying operating system be hidden without limiting the user?

- *An adaptable user interface.* The user interface must provide flexible communication mechanisms to capture and display computer-based design data. The design data is characterized by very complex interrelationships and design domain dependency. Natural language interfaces, context-oriented text search methodologies, and icon-driven user interfaces are emerging software technologies which should be explored for application to the ULCE user interface. Key implementation issues which must be addressed include:

  (1) How can large amounts of complex, interrelated data be effectively captured and displayed?

  (2) Are design application-specific languages [such as the VHSIC Hardware Description Language (VHDL)] appropriate for capturing design knowledge?

  (3) How can graphics be used to access the data base?

  (4) Can adaptable user interfaces which incorporate cognitive models of the various users of the system and learn individual users' styles be developed?

- *Standards for data exchange and software.* Standards are necessary to streamline the inefficient and paper intensive process for exchanging product

data. Mechanisms for exchanging data include hardware standards (i.e., EIA-232C), protocol standards (i.e., MAP/TOP), and data exchange standards (i.e., IGES, EDIF). The ULCE environment will leverage from existing and emerging capabilities. In the area of application-independent product data representation, no comprehensive standard exists. The PDES, Product Data Exchange Specification, effort is aimed at developing a complete, application-independent definition of product data. ULCE requirements should impact the development of PDES. Other outstanding issues to be addressed include:

(1) How can a standard definition of product data to replace the engineering drawing be developed?

(2) What mappings should be developed between established data exchange standards and a standard definition of product data?

(3) What interfaces tailored specifically for the design function being performed are needed to display information contained in the product data model?

(4) How should ULCE requirements impact the development of these standards?

(5) What software standards pertaining to object-oriented languages, analytical software, and operating systems are needed and how should they be developed?

- *Conceptual and product models of engineering design data.* An extensible representation of design engineering data which incorporates features of the network, hierarchical, relational, and object-oriented data model types is needed for ULCE. This data model will provide a foundation for the integration of the ULCE design tools and the knowledge/data base management systems. To provide a standard definition of the design data needed throughout the life-cycle of a product, the following issues must be addressed:

(1) What type of data model is appropriate for the representation of conceptual design engineering and product specific data?

(2) At what level of abstraction should the design data and engineering knowledge be represented?

(3) What methods are needed to define and validate both conceptual and product data models?

- *An advanced data base management system.* The advanced data base management system will control the administration of shared engineering data and the concurrent manipulation of the data by many interactive users. It will organize the engineering data and design knowledge across multiple

representations of the same design (design alternatives) and multidisciplinary functions. Issues which must be addressed include:

(1) How do you manage complexity in a run-time environment?

(2) What data base management tasks need to be performed in an ULCE environment?

(3) What automatic methods are needed to reason with and verify the contents of the data bases?

(4) What methods are needed to store and make use of historical design data (design experience and previous designs)?

- *Mechanisms for implementing the Meta-Design process.* The ULCE architecture design decision process planning approach requires automated tools to model the structure of the decision process and utilize heuristics to determine the extent that a design feature impacts the set of driving requirements. These tools must be fully integrated with the ULCE operating environment, the design representation (the Design-in-Progress), the data base management system, and the automated analysis tools. The manner in which the decision process structure is formulated and manipulated needs further exploration.

E. CONCLUSIONS AND RECOMMENDATIONS

1. General Philosophy of the ULCE Design Process

The need for an ULCE architecture has grown out of the recognition of the need to consider supportability and producibility in earlier design phases and at the same level as performance, cost, and schedule. The potential advantages of this approach are numerous and have been discussed in many places, therefore, they need not be enumerated here. What is important is recognition that reaching this goal involves incorporating many more design considerations into every phase of the design process, leading to increased coordination and communication problems. In many cases, the current design process does not handle the current load of performance, cost, and schedule tradeoffs well.

Therefore, the ULCE problem should be considered one of addressing the bigger picture of reducing the limitations in the current design process which will make consideration of more requirements during all design phases possible. ULCE, therefore, improves the design process for the full range of design considerations including, but not limited to, supportability and producibility. The proposed Meta-Design approach to engineering design does not explicitly show supportability and producibility processes, but

it groups them with the full set of possible requirements. By allowing the design process to be modified based on a program's requirements, the Meta process fully supports incorporation of the desired mix of performance, supportability, producibility, cost, and schedule considerations.

2. Feasibility of Implementation

Two of the three major sections of the ULCE architecture, *Generate Design Alternatives*, and *Make Design Decisions*, are felt to be evolutionary, requiring logical growth of existing or planned approaches. The Meta-Design approach embodied in the *Plan Design Decision Process* is the key revolutionary approach proposed. Central to this approach is the idea that a design process, in particular the decision-making process, should be driven by the peculiar requirements and system definition of a program. This section does not have a close model in today's design environment and, therefore, involves the major implementation issues of the ULCE architecture.

Successful development of the Meta-Design is the central issue of the ULCE architecture. The major underlying issue or enabling technology is the capture of an explicit representation of all design relationships that might affect a specific class of design problems. Because these relationships must be formulated in computerized form, this task is formidable from the software development standpoint. The object-oriented approach appears to make this task feasible. However, several issues require additional development:

- Developing design decision strategies
- Establishing how sensitivities are passed between design tasks
- Scheduling design tasks.

The system must provide a range of decision support tools, including the capability to handle both quantitative and qualitative decisions. Research is underway now in assessing sensitivities but only for quantitative parameters (Sobieski at NASA Langley). While methods for scheduling design tasks is a developmental issue, it is not considered to be a critical problem. A number of approaches can be applied to this task, including knowledge-based methods and methods derived from operations research.

3. Recommendations

Based on the conclusions drawn from the study, the study team makes the following recommendations regarding ULCE research and development, and implementation strategies:

RESEARCH AND DEVELOPMENT ISSUES

- *Additional research on human interactions in design should be conducted.*

  A key element of the ULCE architecture is a team approach to development of design alternatives. This raises some questions on how best to implement such a team concept. Which *ilities* engineers should be on the team, and how many? What kind of techniques could be employed to help a team reach consensus in the face of a large number of design decisions and which factors must be considered? What lessons can be learned from previous team design efforts such as approaches which were successful versus those that were not?

- *Theory, methodology, and tools need to be developed for design process planning, execution, and control.*

  Significant research has been directed at manufacturing process planning, but little at design process planning. Key disciplines involved with this research would include general design theory and methodology, operations research, decision science, management science, and artificial intelligence. Because design process planning must be driven by actual design requirements, methods, and technology, it is clear that to be successful advances in this field must be coupled closely with specific domain knowledge in the various engineering disciplines. Thus, to be successful, work in this area must be multidisciplinary in character.

- *Research should be conducted in the areas of data base management systems, data modeling, and applications of object-oriented technologies to design systems.*

  These fields are key to development of the Design-in-Progress information base which is the foundation for the ULCE architecture described in previous chapters. Research needs to be directed not only to application of object-oriented techniques to the representation of physical design data (parts, sub-assemblies, and assemblies, etc.) but also to the representation of design requirements and functional hierarchies which are needed in systems engineering and to the representation of design tasks in design process planning, analysis, and control.

- *User interface issues are critical in making ULCE a viable environment for design of producible and supportable systems.*

  While object-oriented and symbolic computing technology may be required as the foundation for a mature ULCE system, the designer must be shielded from such technical details and allowed to work in a way which is natural for him. Research is needed on development of user interfaces which will allow designers the freedom to be as creative and productive as possible with a minimum requirement for learning computer specific languages or protocols.

- *Techniques for automated generation of design detail must be further developed.*

  Currently, there are several computer aided engineering systems which provide for parametric syntheses of design geometry. Under the ULCE architecture presented in this paper, there will be a continual requirement for changing designs creating new alternatives, and creating variants of existing alternatives. Without automated tools which quickly develop adequate design detail for analysis of these alternatives, time and manpower will cause the ULCE design process to be prohibitively expensive. Silicon compilers in IC design have already provided a great deal of automation to generate design detail. Similar capabilities must be developed in other engineering disciplines.

IMPLEMENTATION ISSUES

- *There is a critical need for development of a comprehensive, phased plan for development and application of ULCE related technologies and implementation of ULCE supportive practices.*

  This plan must address what is feasible and reasonable for the government to do, what must be accomplished within industry, and what the government can do to stimulate industry to do their part.

- *People issues, such as implementation of team concepts in design of defense systems, can be addressed now and can show significant payoffs with minimal requirements for technology development.*

  Implementation of a team approach to design requires first and foremost a strong top management commitment to do whatever needs to be done to make the approach successful. Secondly, it requires breaking down the barriers of communication between the various specialty communities and the design community within each company (and within the government.)

- *The design community must play a key leadership role in ULCE development and implementation.*

  The job of designer is already difficult when only requirements for performance, cost, and schedule are considered. However, it becomes even more difficult when the great emphasis that is now being placed on requirements for producibility and supportability are added. This problem cannot be solved by *ilities* specialists operating in relative isolation, developing these additional requirements for designers. The only solution is in the designers leading a cooperative effort in developing requirements.

F. NEXT STEPS

As part of this year's (FY88) research program, IDA will be examining various techniques that might be employed in analyzing the relationships between design attributes and requirements, in developing the design process plan, and in measuring and quantifying supportability and producibility of designs in the concept and preliminary design stages. IDA also is conducting research relating to product data definition and exchange standards, and will continue to work in this area. As with the effort presented in this paper, IDA will work cooperatively with defense contractors and academia in conducting this research.

ARCHITECTURE AND INTEGRATION REQUIREMENTS FOR AN ULCE DESIGN ENVIRONMENT

A. INTRODUCTION

This report is the result of the work performed by the Institute for Defense Analyses (IDA) under Task Order MDA 903 84C 0031, Task No. T-B6-508, "Architecture and Integration Requirements for an ULCE Design Environment." The work was performed for the Air Force Human Resources Laboratory and the Under Secretary of Defense for Acquisition (USDA). A major contributor to the report was Lockheed-Georgia who provided the results of their effort under an IDA subcontract in support of the Architecture and Integration task.

1. Background

The Architecture and Integration task is just one of a number of Unified Life Cycle Engineering (ULCE) Program activities. The stated objective of the ULCE Program is "...to develop, demonstrate, and transfer to application the techniques and technologies needed to provide advantageous computerized integration of the procedures dealing with designing for producibility and supportability with those dealing with designing for performance, cost, and scheduling..." [Ref. 1]

The need for improvement and changes in the design process which will allow or even force early consideration of factors related to producibility and supportability in addition to the usual factors of performance, cost, and schedule is implicit in that objective. This need for improvement also relates to both timeliness and prioritization of the various design requirements.

A second need is a new family of design tools to aid in formulating measurable requirements for producibility and supportability and to support the design process.

It must be remembered that the new design processes and design tools need not be fully automated. In all probability, if a structured approach is taken to the development and

implementation of ULCE, some of the required changes and new developments will evolve through non-automated but, of necessity, integrated stages. There have been a number of very successful efforts, principally in the commercial product world, where the design process was radically changed to incorporate producibility and supportability up front in the process [Ref. 2]. Although in all cases CAE/CAD tools were used by the designers, they were not truly integrated. The major changes were in the early introduction of producibility and supportability disciplines and the leverage given those requirements in the overall process.

2. Task Objective

The IDA task objective was to examine alternative architectures for the design process needed to implement ULCE and analyze the problems of assimilation, interpretation, and integration of diverse data bases and analytical tools which must be solved to implement such architectures. More specifically the effort was to:

(1) Investigate architectures for the design process which address multiple levels of subsystem hierarchy and which incorporate concurrency when considering factors related to producibility, supportability, performance, cost, and schedule in the design process.

(2) Develop a set of requirements and specifications for a conceptual design of an executive and a control system to implement ULCE architecture.

(3) Identify trends in research relevant to integration problems which will impact ULCE systems development.

3. Some Aspects of an ULCE System

Unified Life Cycle Engineering (ULCE) has been defined [Ref. 1] as a design engineering environment. The design tools required to meet the needs of that environment can be considered as ULCE systems. An ULCE system will be a computerized integrated design process incorporating design procedures, data and knowledge bases, analysis, simulation, and optimization routines. A key element will be the capability to perform the iterative analysis-design feedback cycles involving all of the design requirements and design parameters across all levels of a hierarchical design structure.

Considering the complexity of the overall design process for even relatively simple products or equipment, it is clear that an ULCE system will, of necessity, be composed of a number of inter-dependent subsystems, and because it will truly be a system, it is critical

that system engineering techniques and disciplines be fully applied in the development of any ULCE system.

4. Scope of ULCE

The sequence of steps or stages in the life cycle of almost any product or equipment can be represented by the simplified diagram in Figure A-1. The point at which the design process begins will depend on the viewpoint and responsibility of the designer. It is clear that decisions or choices made during concept definition can have a major impact on the performance, producibility, and supportability of the end product, and decisions and choices at that stage are part of the design process.

The heirarchical nature of the design process also is evident in Figure A-1: the decomposition (and subsequent reassembly) of the system into smaller functional or physical elements. This hierarchy also can be represented in the vertical dimension as in Figure A-2.

An indication of the complexity of the process is the flow diagram in Figure A-3 which represents only a portion of the detailed design phase at the element level, in this case an electronic printed circuit board [Ref. 3]. The diagram contains only those activities addressing reliability (R) and maintainability (M) and does not include design activities for basic performance, producibility, or other aspects of supportability (besides R&M). An ULCE system for designing at the printed circuit board level and for incorporating the full range of ULCE requirements will almost certainly be a stand-alone design tool that must interface easily with the next higher level.

5. ULCE System Considerations

The overall design process is naturally hierarchical. Furthermore, it is routinely decomposed into modular elements that match the capabilities and resources of the designer and the organization. As a result, ULCE systems, at least in the early stages of development, can be expected to be hierarchical and modular. However, it is critically important that although different levels of ULCE systems may be modular, the resulting product design must not be modular (i.e., the ULCE requirements and design parameters must be transported and evaluated throughout the design hierarchy and at all phases of design).

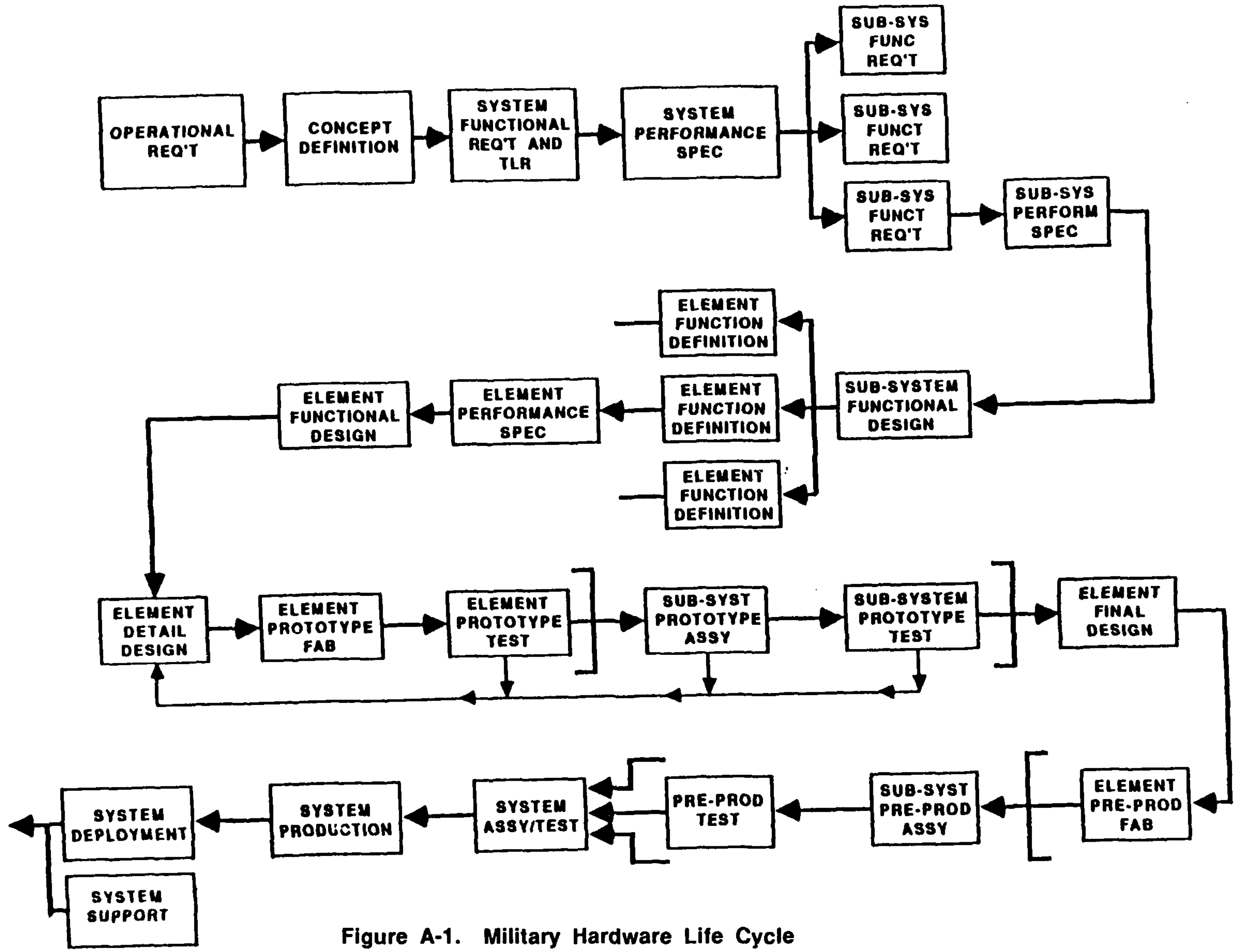


Figure A-1. Military Hardware Life Cycle

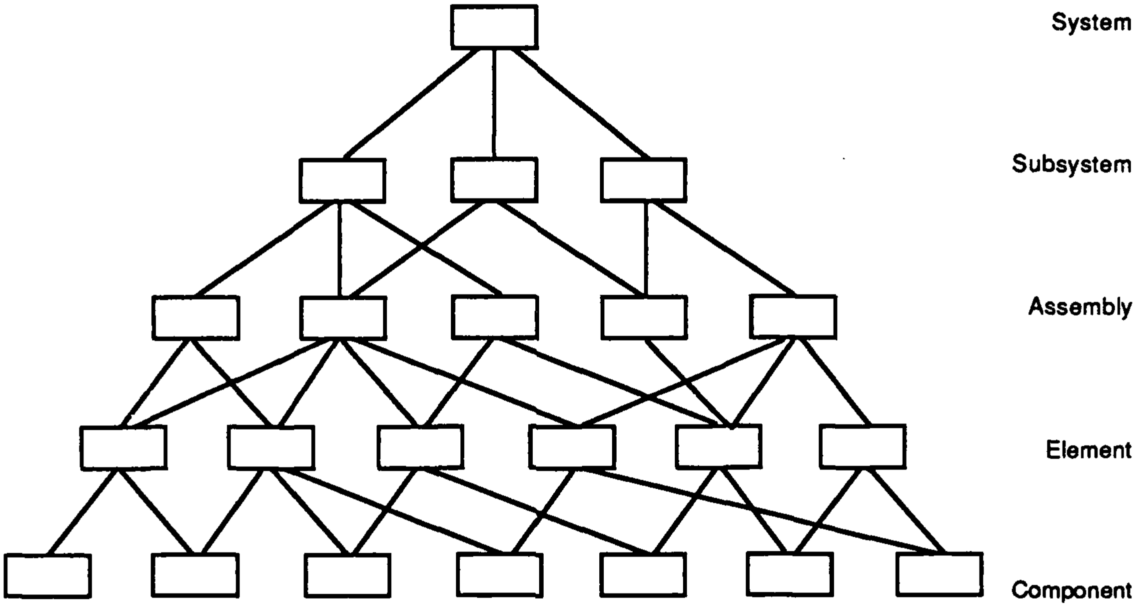


Figure A-2. Functional Hierarchy

EXAMPLE IS FOR THE SUBDIVISION OF A WRA (LRU)

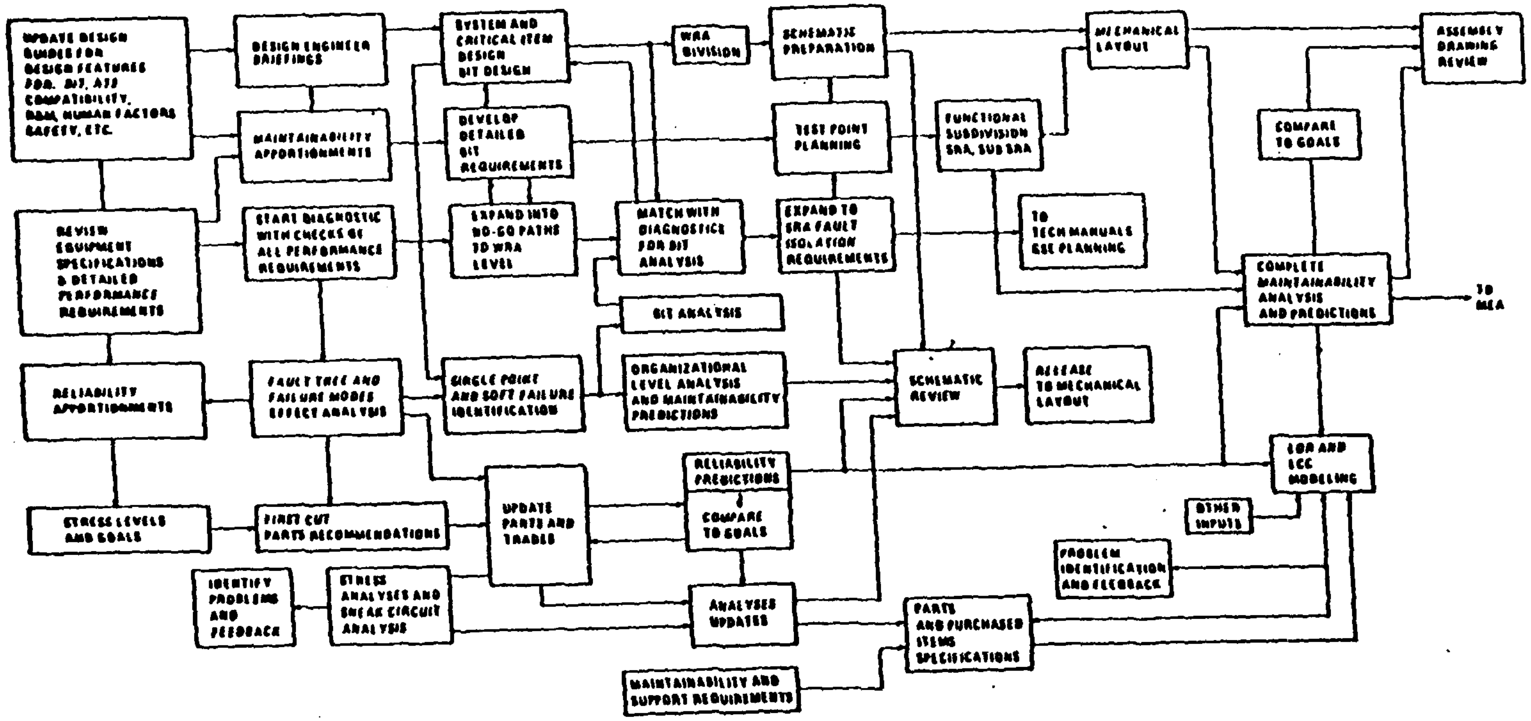


Figure A-3. Typical Reliability/Maintainability Analyses-Design Function

6. Task Approach and Report Organization

a. Task Approach

It is obvious that it is beyond the scope of this study to attempt to define a complete ULCE architecture in enough detail that it covers the total design process for even a very simple product or piece of equipment. It was recognized early in the study that if the effort were not focused on a specific or limited design task, the results would almost certainly be too general to be of use in pointing the way for new development efforts.

Even with that caveat no attempt was made to define a detailed ULCE architecture or to structure a software approach to ULCE. The effort was directed at identifying major shortcomings of the current design process, formulating the concepts for a new or modified process that would effect the principal ULCE objectives, selecting a technological approach to implementing the ULCE architecture, and, then, identifying the requirements needed to support that architecture.

In surveying CAE/CAD in the electronic, mechanical, and structural design areas, it appeared that the state-of-the-art was more advanced and the pace of new developments more rapid in the field of electronics, particularly from the standpoint of ULCE integration objectives. It is becoming commonplace to integrate circuit simulations, test routines, reliability analyses, and cost estimating routines into a single CAE/CAD program. Although large numbers of CAE/CAD programs exist in the mechanical and structural design areas, they usually are limited to single function programs.

Because there appears to be a higher level of automated/integrated design activity in the electronics field, an early decision was made to concentrate the effort on the structural and/or mechanical design process.

Discussions were held with a number of universities and commercial organizations to evaluate independent research and development activities that were closely allied to ULCE objectives. Encouraged by and subsequent to those discussions, IDA issued a Request for Proposal to a number of organizations for support of the ULCE Architecture and Integration Study.

Lockheed Aeronautical Systems Company was selected to participate in the study. There are two aspects of ongoing Lockheed activities that are particularly germaine to ULCE objectives.

- They have instituted a long-term planning approach to integrating computer-aided tools for all engineering activities.
- They have recognized the need to provide concrete measures of supportability and are developing Supportability Design to Requirements (SDTRs) to be integrated at early stages of the design process.

Both of these activities provided a base for Lockheed's efforts in the study.

There were a number of different systems proposed as candidates for evaluation of the current design process and application of the ULCE architecture. However, based on the earlier discussions relating to development activity in the electronic, mechanical, and structural design arenas, a combination of a mechanical/structural design was chosen, i.e., a landing gear design. Specifically the High Sink Rate (HSR) Landing Gear for the C-130 aircraft was used as a model in the study of the ULCE architecture.

b. Report Organization

This report is organized along the lines followed by the study and is directly correlated with the subtasks contained in the Lockheed Statement of Work (SOW).

Section B, System Description, targets the high sink rate (HSR) landing gear on the C-130 aircraft as the design system to be studied.

Section C, Current Design Process, addresses the design process for any landing gear. (There is nothing unique about the HSR gear that would modify the design process.)

Section D, Critical Design Goals and Requirements, addresses landing gear requirements and some of the tradeoff aspects in the design process.

Section E, ULCE Architecture, contains a description of the three major elements that make up the proposed ULCE Design Process. It also addresses the technology approach to implementation of the proposed architecture.

Section F, Software/Hardware Requirements, addresses both the need for expansion/improvement of existing CAE/CAD tools and new tools yet to be developed.

Section G, Evaluation of Proposed Architecture, includes feasibility, design quality improvements, cost of implementation, and life cycle cost aspects.

Section H, Summary and Recommendations, addresses the key points of the study and identifies a number of areas that should receive early R&D attention.

B. TARGET SYSTEM DESCRIPTION

This Architecture and Integration Study for Unified Life Cycle Engineering examines the design of replacement main and nose landing gears for the C-130 aircraft. Unified Life Cycle Engineering (ULCE) involves the total integration of producibility and supportability into the design process. The landing gear offers several advantages as the target system for the study. The landing gear is critical to the safety of flight. While the primary function of the landing gear (to absorb energy during landing) is designed to "safe life" criteria, maintenance and other supportability considerations play a significant role in the design process. Tire selection, jacking considerations, and operation in sand, mud, and dirt environments have a strong impact on supportability.

The landing gear design process illustrates the impact of changing requirements on the design. The need for a new C-130 landing gear is based on AirLand Battle 2000 and R&M 2000 needs to operate from dispersed locations having rough, short fields. The C-130 will be required to land at 15 feet per second sink rate at 130,000 lbs. The original requirement is a 9 feet per second sink rate. In addition, the new C-130 gear must be ruggedized for rough field operations the spare parts and preventive maintenance must be minimized; the aircraft must be maintainable in a nuclear, biological, and chemical (NBC) environment; and the rapid repair, without support equipment, facilities, or skilled personnel, must be accomplished to reduce maintenance manhours.

The landing gear design process also illustrates the interaction between prime airframe contractors and vendors. A high sink rate landing gear has been designed, built, drop tested, and installed on Lockheed's C-130 High Technology Test Bed (HTTB). Five levels of hierarchical breakdown were considered in the design of the high sink rate gear. A detail of the drawing breakdown for the high sink rate gear is shown in Figure B-1. Each block in the Figure B-1 hierarchy represents a configuration item that is controlled by a drawing in the drawing breakdown. The HTTB aircraft is at the top level. Four subsystems of the HTTB are referenced at the next level down (second level). These are: the nose landing gear, main landing gear, airframe primary structure, and the aircraft hydraulic system. It should be noted that the design of a landing gear for all-new aircraft would involve more interactions with the aircraft electrical system and other systems.

The nose landing gear (NLG) subsystem is illustrated at the third level of the design hierarchy. The third level elements of the nose landing gear subsystem include the shock strut, steering cylinder, uplocks and restraints, brakes, tires, and wheels. The NLG shock

strut is described at the fourth level and includes components such as cylinder, piston, bearings, tow arms, jack pad, contraction piston, etc. Individual parts of the contraction piston assembly are identified on the fifth level of design hierarchy and include the contraction piston, lock pin, and spacer.

The installation of the high sink rate landing gear on the C-130 HTTB is shown in the Figure B-2 sketch. This sketch also indicates the impact of the 15 feet per second sink rate requirement on the C-130 primary structure.

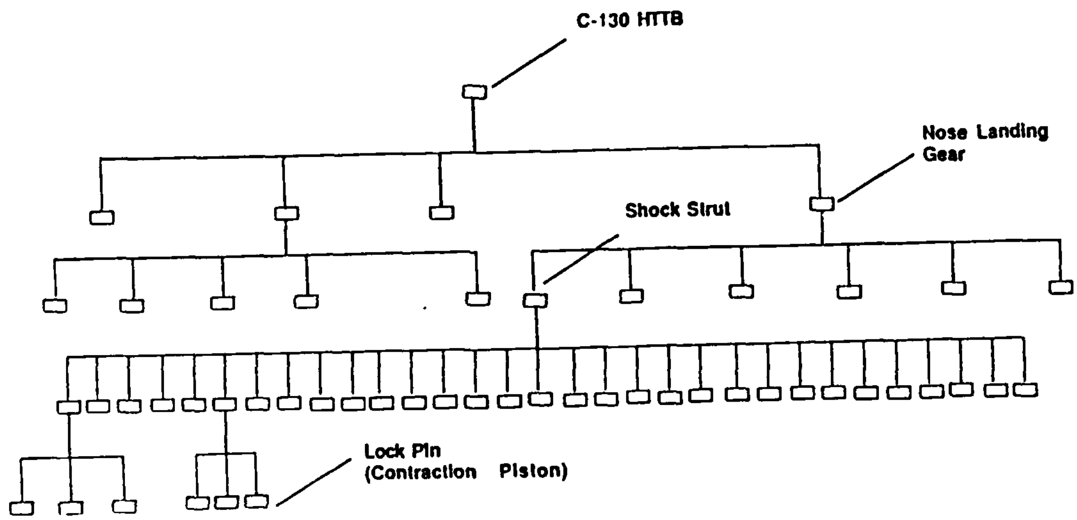


Figure B-1. High Sink Rate Gear

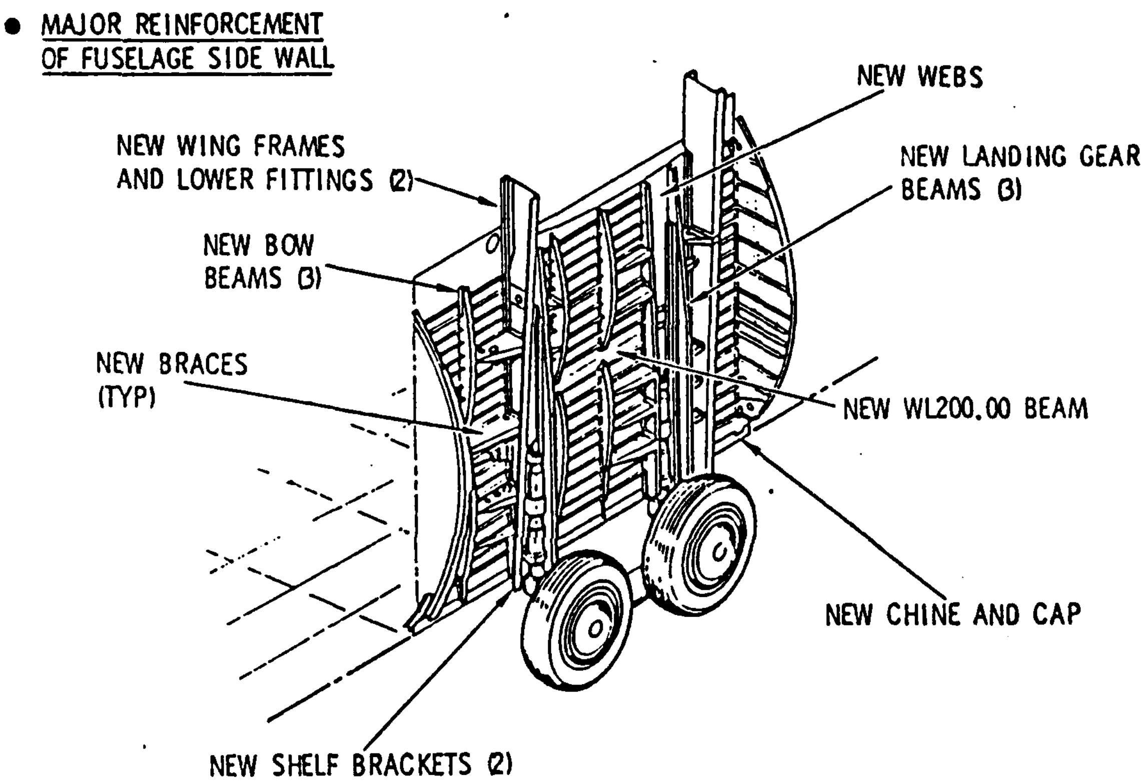


Figure B-2. High Sink Rate Landing Gear on the C-130 HTTB

C. CURRENT DESIGN PROCESS

1. Overview

The current design process emphasizes the build-up of design definition. The central architectural feature of the current process is the iteration between design definition and design review (Figure C-1). The technology for developing design definition is highly labor-intensive. Sketches, drawings, engineering drawings, etc., require a considerable investment to prepare and modify whether this is performed manually or using computer-aided two-dimensional or three-dimensional drafting packages. As a result, current engineering practice in landing gear design is characterized by a relatively rigid sequence of design decisions.

The architecture of procedures in the current design process (Figure C-1) is elemental, rather than top-level. The next level of detail in describing the current process is constructed by chaining elements (such as the one shown in Figure C-1) together to address each design decision in sequence, as shown in Figure C-2. The sequence is not currently based on a prioritization of requirements, but is driven by the costs associated with iterative redesign loops as in Figure C-2, the details of which are shown in Figure C-3. The sequence of design decisions reflects the need to push design decisions requiring considerable design definition to the end of the sequence.

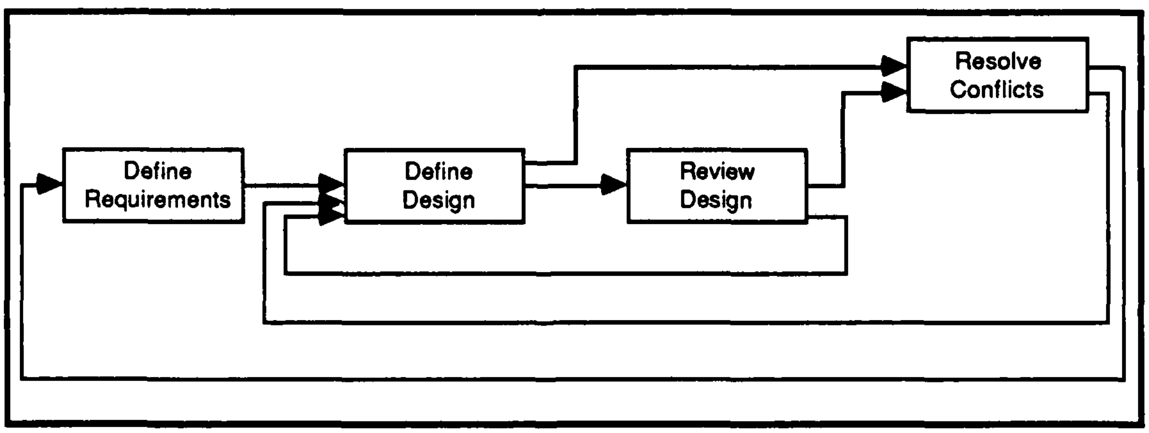


Figure C-1. Iteration in Current Design Process

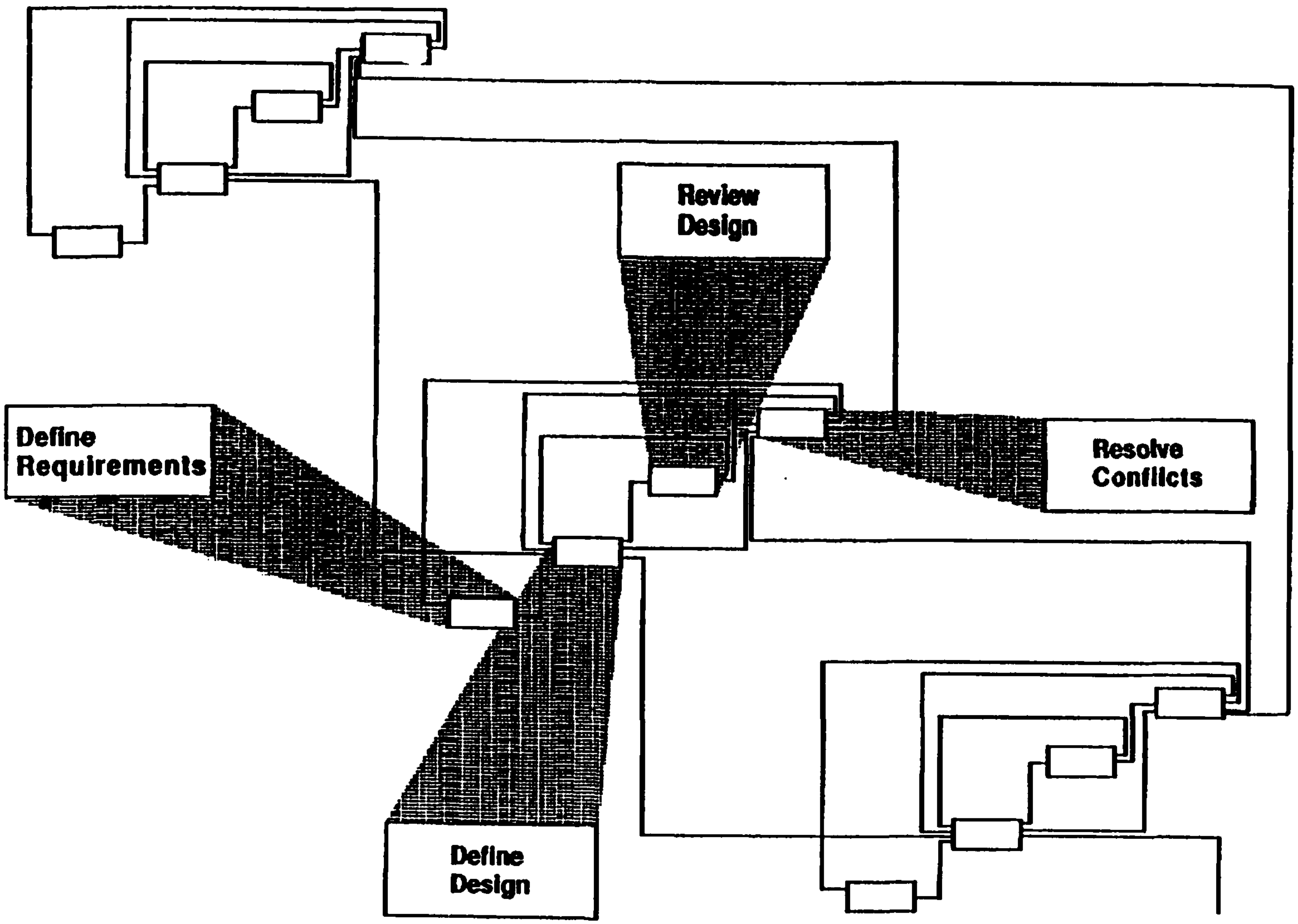


Figure C-2. Iterative Redesign Loops in Design Decision Sequence

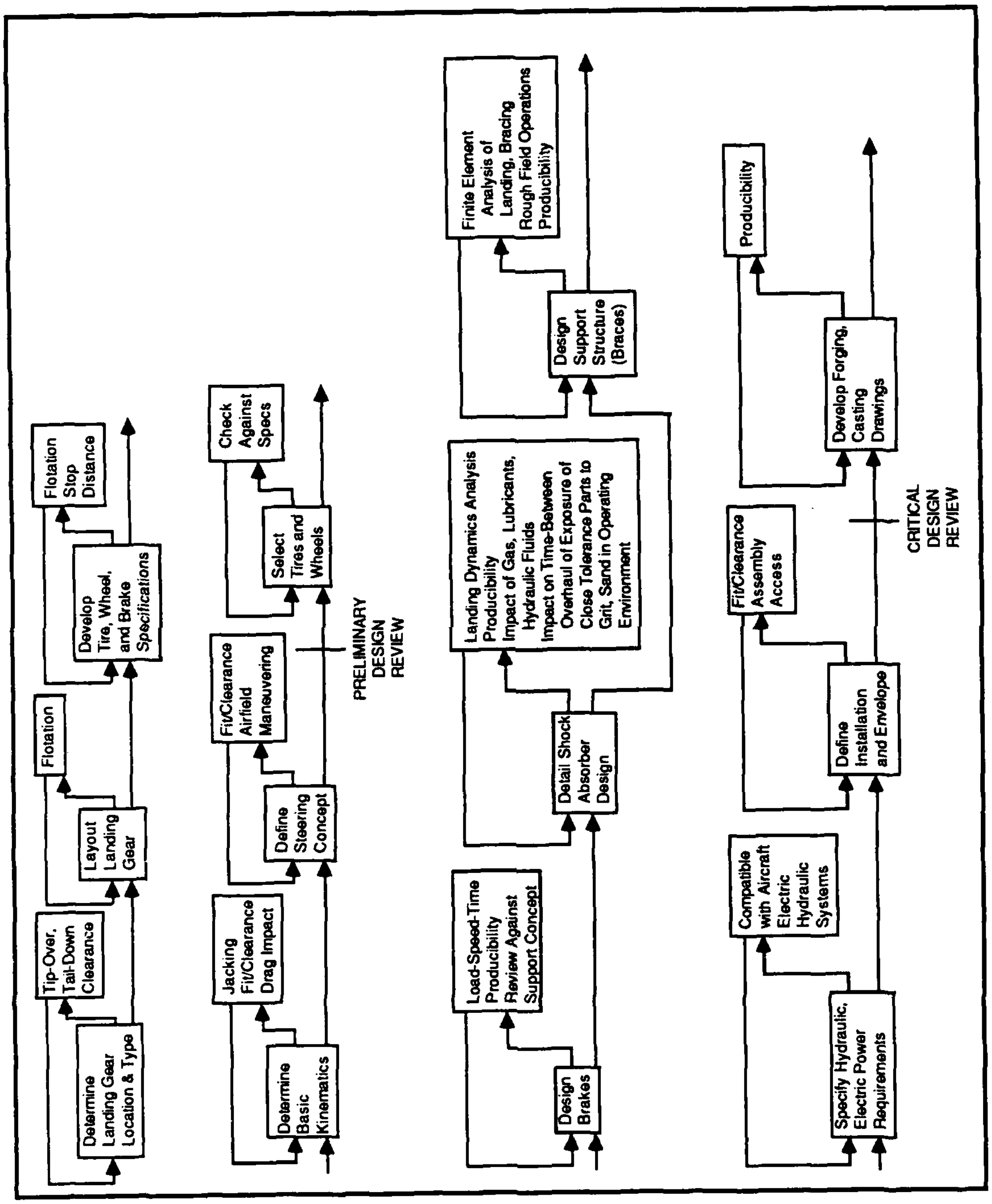


Figure C-3. Details of Iterative Loops

Producibility, supportability, and all other engineering specialties are currently incorporated into the design through the requirements definition and design review elements of the procedural architecture. Requirements definition currently involves all engineering disciplines in deriving requirements implied by the request for proposal (RFP), or based on "lessons learned," operational concepts, and other sources. The role of the landing gear designer (the engineer who performs the design definition task) is to understand these requirements and to translate them into functional and implementation descriptions of design concepts and detailed designs that meet the requirements. If this is not possible, the landing gear designer must be able to clearly draw out conflicts between the requirements. Trade studies are then performed to support a management decision process to resolve those conflicts. The requirements definition process may involve the development and critique of design concepts by the engineering specialists in the course of identifying design-to requirements. *The context provided by these design concepts for understanding the requirement is removed from the statement of the requirement as it is currently presented to the designer.*

Design definition is the development of functions that can accomplish the stated requirements and the invention of devices to implement these functions. The design description is reviewed by a team of engineering specialists, designers working on closely coupled system elements, and technical managers. The description of the design, often graphical, is "hung on the wall" and the designer then explains the features of the design, why they were included, and how this design will perform the required functions. The designer attempts to convince the engineering specialists that the design meets their defined requirements and the RFP requirements.

Several suggestions for design changes will emerge from the review. These are documented in internal correspondence summarizing the design review meeting minutes. The landing gear designer then attempts to integrate these changes into the design. Should conflicts arise that cannot be resolved by the designer (individual decision making), or by the design review process (group decision making), the need for a technical management decision is identified.

2. Procedure Flow

An overview of the flow of procedures through the landing gear design process, from concept development through Critical Design Review (CDR), is shown in Figures C-4 and C-5 (Ref. 4).

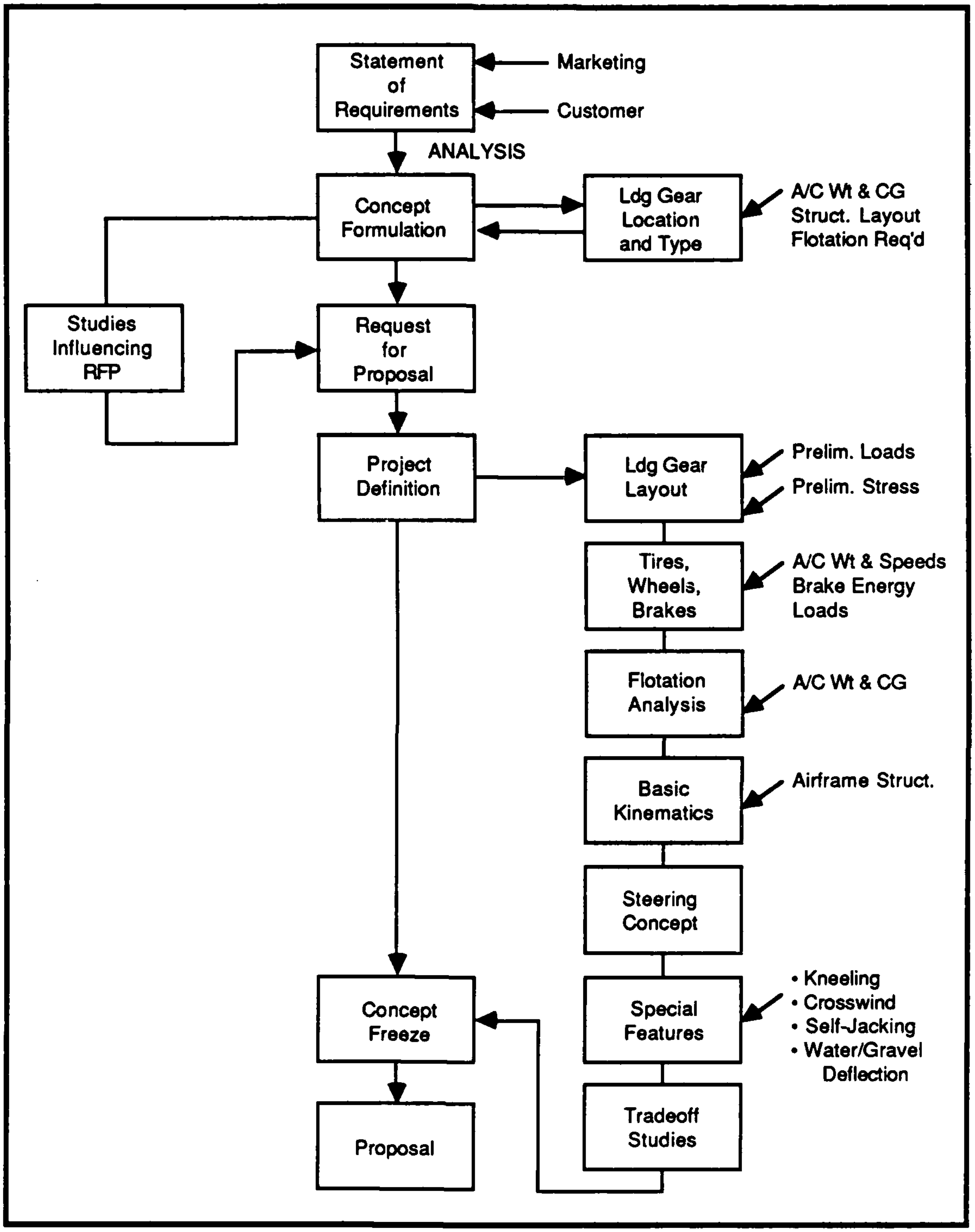


Figure C-4. Landing Gear Preliminary Design

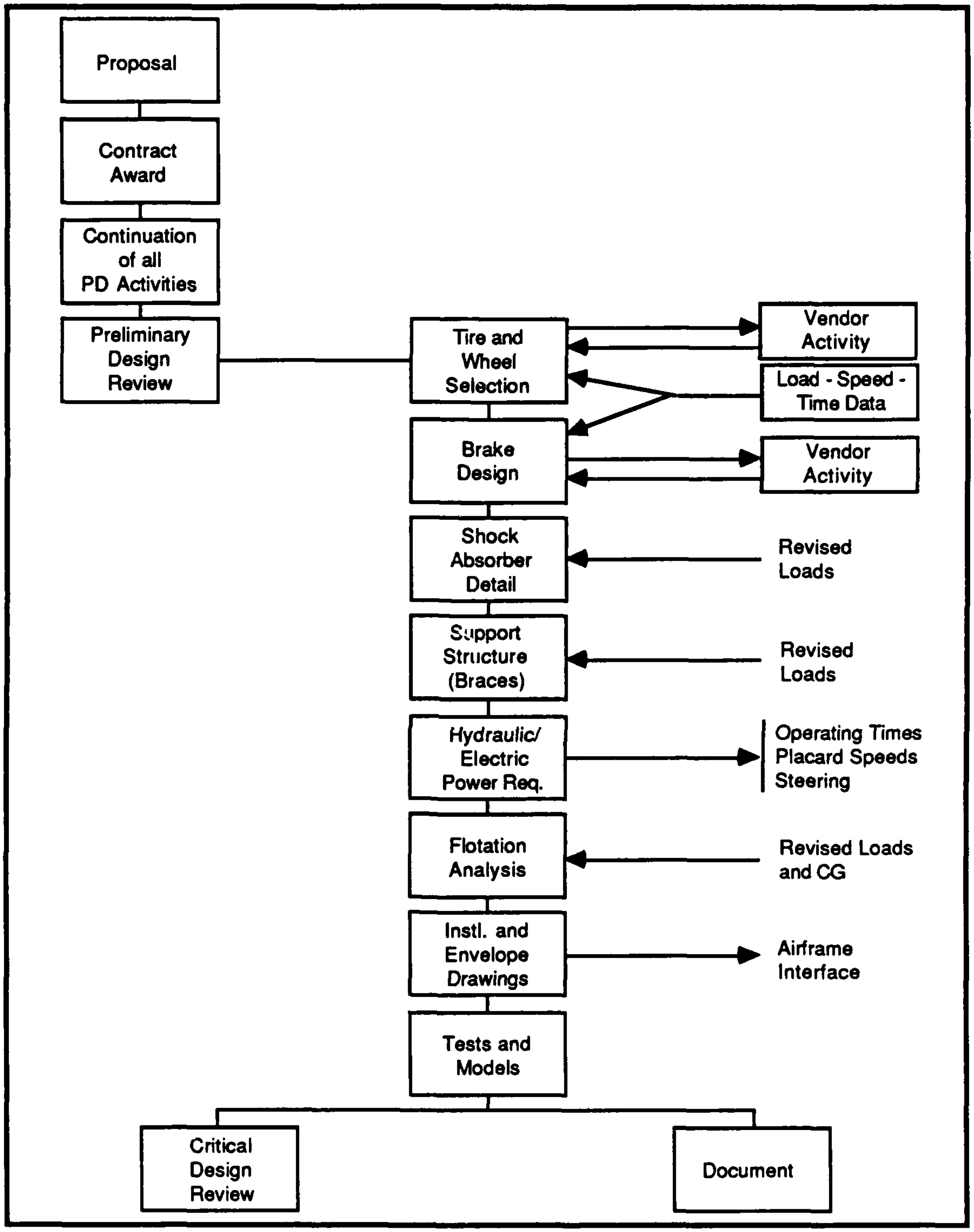


Figure C-5. Landing Gear Post-Contract Design

a. Preliminary Design

The objectives in the preliminary design phase are defined by the concept formulation phase and the project definition phase.

In the concept formulation phase, objectives can be summarized as follows:

- to determine landing gear location as a function of the location of the center of gravity and the general structural arrangement of the airframe
- to determine the number and size of the tires as a function of the weight of the aircraft, the braking requirements, and, if specified, the flotation requirements.

In the project definition phase, objectives can be summarized as follows:

- to decide on the general configuration of the aircraft to allow for a more detailed and analytical design study
- to prepare the proposal with as much detail and credibility as possible

(1) Concept Formulation Phase

An overview of the Concept Formulation Phase is shown in Figure C-5. In this phase there will probably be a number of widely varying aircraft concepts, and only a brief analysis is required for each one. As a minimum, the designer must know the aircraft weight and its range of center of gravity (CG) position. From this, the options for wheel number and size can be determined. For example, one choice that can be made at this stage is the use of two large tires or four smaller tires at the end of a shock strut.

The options are reviewed to see how they match the airframe structure and the flotation requirements (if any). Landing gear location and length are determined by the CG location, by the tail-down angle requirements to suit takeoff and landing attitudes, by the tipover, and by the general airframe configuration. Flotation is checked for the various wheel sizes, using rigid, flexible, and bare soil rules as applicable. A small tradeoff study inevitably results in order to determine the most cost-effective arrangement.

(2) Project Definition Phase

In the Project Definition Phase, there is an urgency to quickly freeze the design concept. The best overall aircraft concept is selected, and the landing gear design becomes more detailed. The continuing aircraft weight, the CG analysis, and the subsequent loads derivation allows the designer to refine gear location and gear loads. Based on the defined sink rates, the approximate strokes are determined at the main gear and nose gear, and the

landing gear dimensions and sizes are established from a rough layout. A layout is then prepared to evaluate and, in particular, to document the taildown angles, turnover angle, and the clearances to deflected surfaces, engine nacelles, and propellers (if used), with various conditions of strut and tire inflation/deflation.

Tire, wheel, and brake vendors are brought into the plant at this point to discuss availability of existing equipment. It is possible that a new tire could be developed for the aircraft or that plies could be added to an existing tire, and these may be a subject of vendor negotiation. The matching of tire and wheel size to brake size is another important activity. To adequately address this subject, the takeoff load-speed-time data, the dynamic taxi loads and landing loads, and the takeoff speed profile used for brake kinetic energy calculations should be available. The relative size, cost, and weight of steel and carbon brakes would be evaluated at this time.

As tire sizes, wheel arrangements, loads, and CG range become defined, the flotation calculations are recycled. Also, airfield roughness requirements would be evaluated at this time. The basic kinematics analysis of the landing gear demands a great deal of ingenuity on the part of the designer. This analysis involves the retraction, extension, and locking systems, with due consideration given to emergency conditions, including free-fall. A wide variety of possible systems can be considered, ranging from those with simple up-and-down motion to those that rotate the entire strut about its axis while properly positioning the bogie at the same time. The objective in all cases is to be able to retract the gear into a cavity that has the least effect on the basic airframe structure and to minimize any external contour changes that would increase aircraft drag. The steering concept is a fundamental part of the nosegear design and must be determined before the proposal is prepared.

Special requirements that may have to be considered include kneeling, crosswind positioning, self-jacking, and deflection of water or gravel. A number of tradeoff studies are conducted in the Project Definition phase. These studies are fully documented and kept on file.

By definition, the preliminary design phase continues until the Preliminary Design Review (PDR) has been completed, although at this time the personnel involved in the design team may well have changed to those who are more oriented toward the project design activity (see Section C.3.b. below). These are the engineers who are better acquainted with design details such as tolerances, surface finishes, current fastener types,

and anti-corrosive measures. For military aircraft, the PDR must be scheduled prior to manufacturing parts. The engineers describe the design to the customer using sketches, block diagrams, concept drawings, and informal documentation, and the customer determines whether the design meets the specification requirements.

b. Post Contractual Design

From the PDR until the Critical Design Review (CDR) the design is refined in every detail so that the design can be finalized and the parts manufactured. A diagram of the work involved is shown in Figure C-5.

Prior to the CDR, the following tasks are performed.

- Tire and wheel selection or design is concluded, load-speed-time data is revised, and vendors are established.
- Brake energy requirements are updated, vendors are selected, and the design is finalized.
- Shock absorber details and support structure are sized to be compatible with the revised loads.
- Electric and hydraulic power requirements are defined for retraction, extension, and steering. Operating times, placard speeds, steering angle and steering rate are determined and turning diagrams are prepared.
- Flotation analyses are updated again to reflect changes in loading on the landing gear.
- Installation and space envelope drawings are prepared to facilitate determination of stowed landing gear clearances, and to provide appropriate information to the airframe designers. This is a primary item for inclusion in the aircraft Basic Data Book that is in preparation at this time.
- Tests and models may be used in this phase to acquire confidence in the proposed design, to gain a better understanding of problem areas, to display complex kinematics, and to evaluate locking mechanisms.

The entire design is then documented for presentation at the CDR.

The detail design and manufacture of the landing gear (or parts thereof) may be subcontracted to one of several companies that specialize in those parts. This practice varies considerably--some aircraft companies design and build their own gears, some design the gears and have the shock struts built by a specialist company, some ask these companies to undertake all of the detail design and manufacture, and some bring in

specialists at the Project Definition Phase. Typical examples of these specialist companies are Cleveland Pneumatic Co. and Menasco in the United States, Dowty in England, and Messier-Hispano-Bugatti in France.

After the CDR, the focus is on detail design of the parts for production, system schematics, system installations, assembly drawings, installation drawings, loads analysis, power analysis (hydraulic and electrical), tests, and procurement activity. Producibility enters the design process as shown in Figure C-6. Forging and casting drawings are usually completed first because of long lead times. Working mockups (full scale) are sometimes employed to prove the kinematics and structural clearances and to facilitate hydraulic routing. Analyses are conducted to evaluate shimmy, dynamic response to airfield roughness, fatigue, and damage tolerance.

Various tests are conducted before first flight. During the design phase, photoelastic tests are often used to show areas of high stress concentration. Static structural tests measure the deflections and spring rate of the gear under load and also confirm its structural integrity. Drop tests are employed to verify shock absorber efficiency and if necessary, to modify metering pin/orifice sizes to improve that efficiency. Shock strut proof pressure and leak tests are conducted, and overall fit, function, and endurance tests are made.

Procurement activity involves such items as wheels, tires, brakes, skid control, actuators, miscellaneous valves and fittings, position switches, and the basic landing gears if they are being designed and/or built by a subcontractor. The normal procedure is the preparation of specifications and vendor drawings from which competing vendors can respond. These responses are then analyzed and rated to select vendors, who, in many cases, must then provide Qualification Test Procedures for approval by the airframe manufacturer. When the parts have been built, they are tested by the vendor who then submits a Qualification Test Report for approval. This ensures that all of the contractor-specified requirements have been met and that full documentation is available to prove it.

Other analyses that are completed before first flight are the Failure Mode, Effect, and Criticality Analysis (FMECA), and the supportability analyses. The FMECA is particularly important because it evaluates the effects of failure on any part in the overall landing gear system and determines its effect on the aircraft. Since this analysis may uncover some deficiencies that had been overlooked, its timing is such that design changes can be made without affecting the first flight schedule.

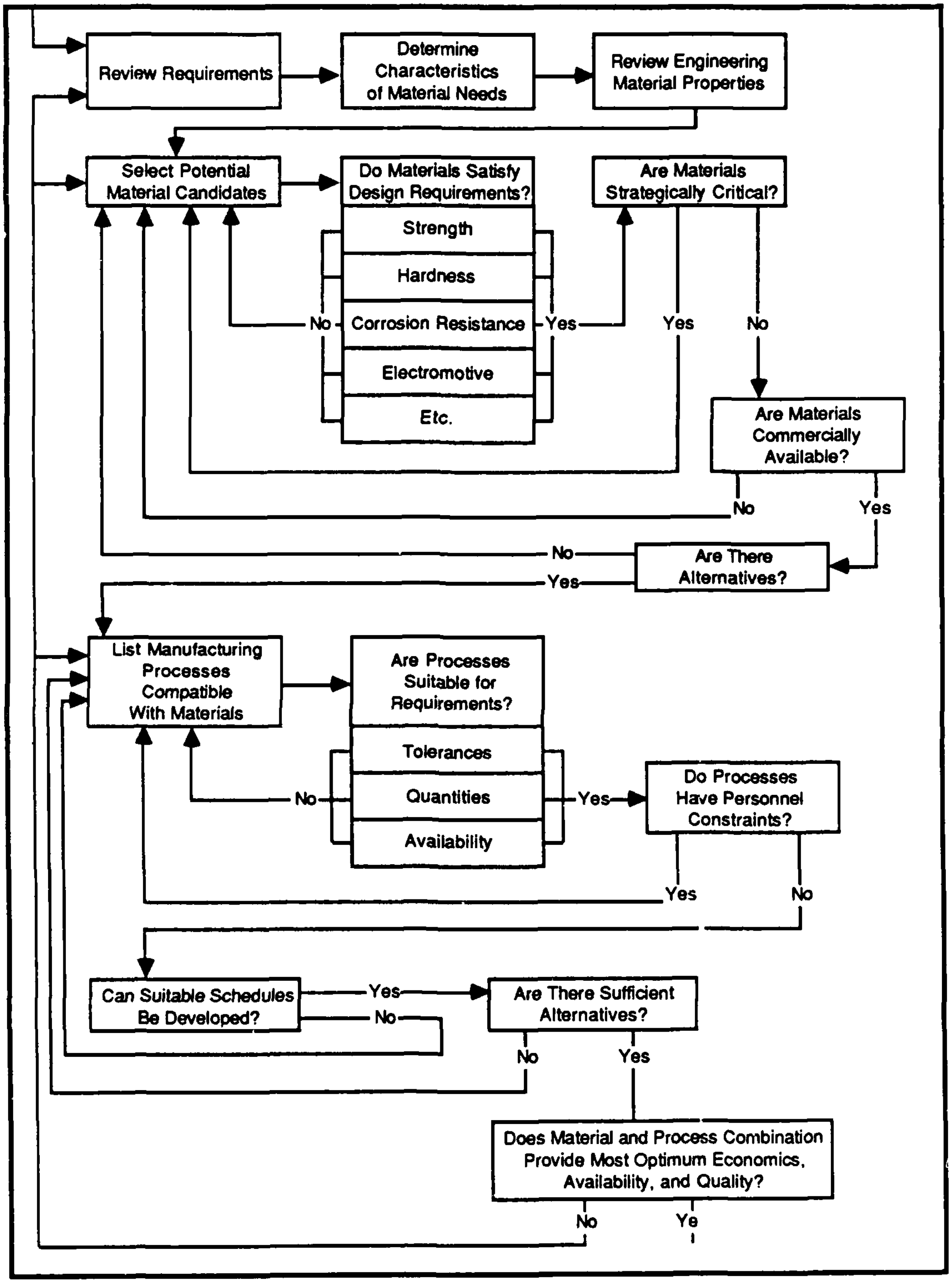


Figure C-6. Producibility in the Design Process

Supportability analyses are performed in recognition of a growing demand for increased mission readiness and improved economics. Figure C-7 presents a simplified form of the supportability design process. As indicated by the chart, system criteria are the result of performing the logistics support analysis (LSA) and serves as the method to implement the technical strategy indicated by the LSA data. That is, the LSA identifies the system/equipment needs, the technical strategy is a chosen way to achieve the needs, and the system criteria are the necessary features and design characteristics to fulfill the needs.

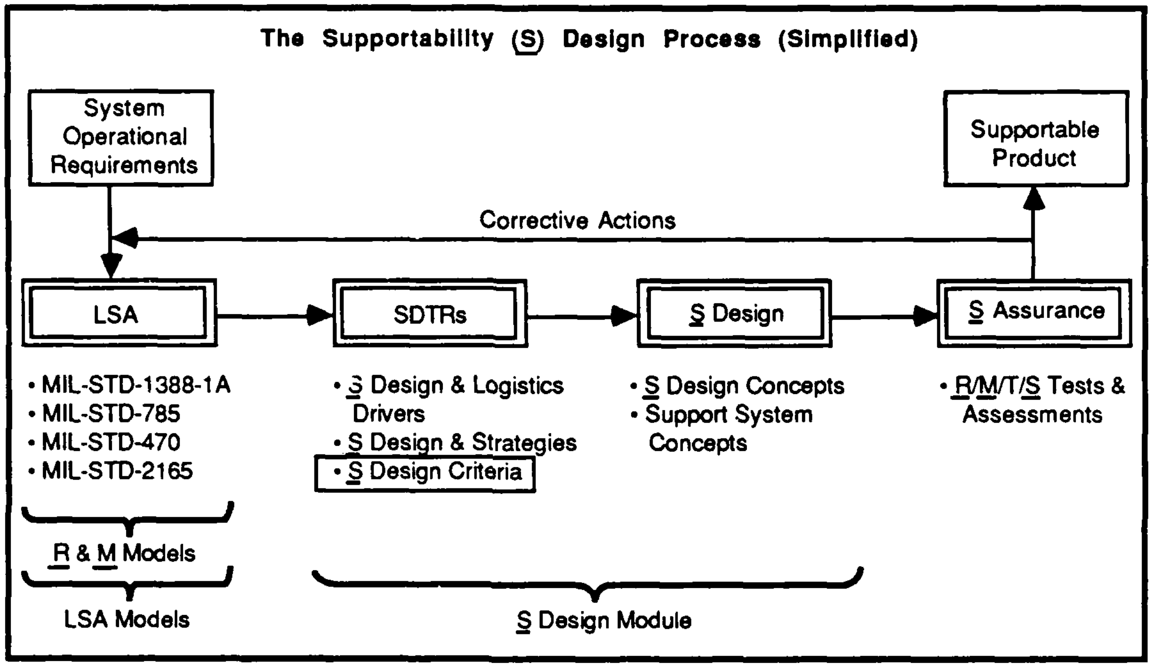


Figure C-7. Simplified Supportability Design Process

The time between overhaul (TBO) must be estimated by reviewing data for similar equipment operating in a similar environment, and appropriate changes must be legislated for aircraft operating in adverse conditions. Using the facilities at the manufacturing plant, field reports, and "lead the fleet" aircraft, an early determination can be made of critical areas. If any of these areas require overhaul prior to the airframe TBO, then they must be corrected so that landing gear and airframe TBO coincide. Similarly, if it appears that minor gear modifications will stretch its TBO to double that of the airframe, they should be considered. To a great extent, the landing gear TBO is, therefore, an arbitrary decision based on spares availability, economics, and convenience.

3. Information Flow

Throughout the entire design process, from the development of first concepts through production configurations, it is extremely important that complete documentation be maintained. For each aircraft configuration there is a listing of its assumed weights and geometric data in the landing gear files. The designer has a summary attached to each file to show the basic essentials of the gear. The depth of the summary depends on the seriousness of that particular configuration and/or the complexity or uniqueness of the landing gear involved.

Details of the data flow for the Preliminary and Post Contractual Design Phases are shown in Figures C-8 and C-9.

4. Organization

Details of the organizational structure of the Lockheed Aeronautical Systems Company that pertain to the design for performance, producibility, and support of landing gears are shown in Figure C-10.

a. Structure

Conceptual and preliminary design of landing gears is done in the Aeronautical Systems Development Department of the Preliminary Design Division, reporting to the Chief Research and Technology Engineer. The detailed design of landing gears is done in the Hydraulic and Controls Design Department which is a part of the Mechanical Division of the Chief Project Engineer's office.

(1) Supportability

All engineering activities related to product support are under the direction of the Chief Engineer--Supportability, reporting directly to the Vice President--Engineering. The supportability organization consists of the Advanced Supportability Engineering Department, the Supportability Engineering Division, and the Engincering Service Publications Division.

Supportability design functions performed at the Lockheed Aeronautical Systems Company are Logistics Support Analysis (LSA), Maintainability (M), and Reliability (R), which are described by MIL-STD-1388-1A, MIL-STD-470A, and MIL-STD-785B, respectively. Supportability design-related functions performed by the organization are logistics records development defined by MIL-STD-1388-2A; research associated with

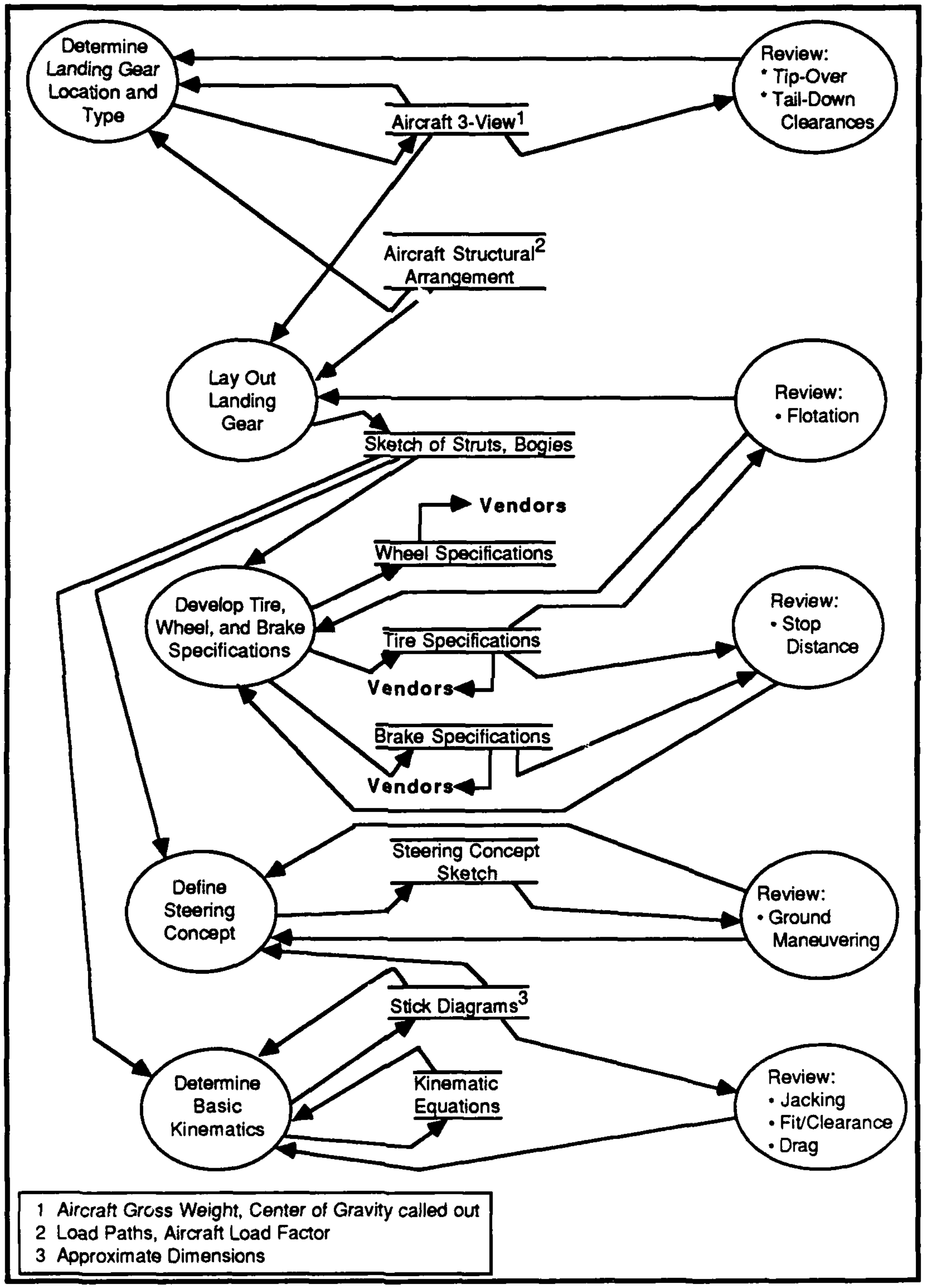


Figure C-8. Preliminary Design Phase Data Flow

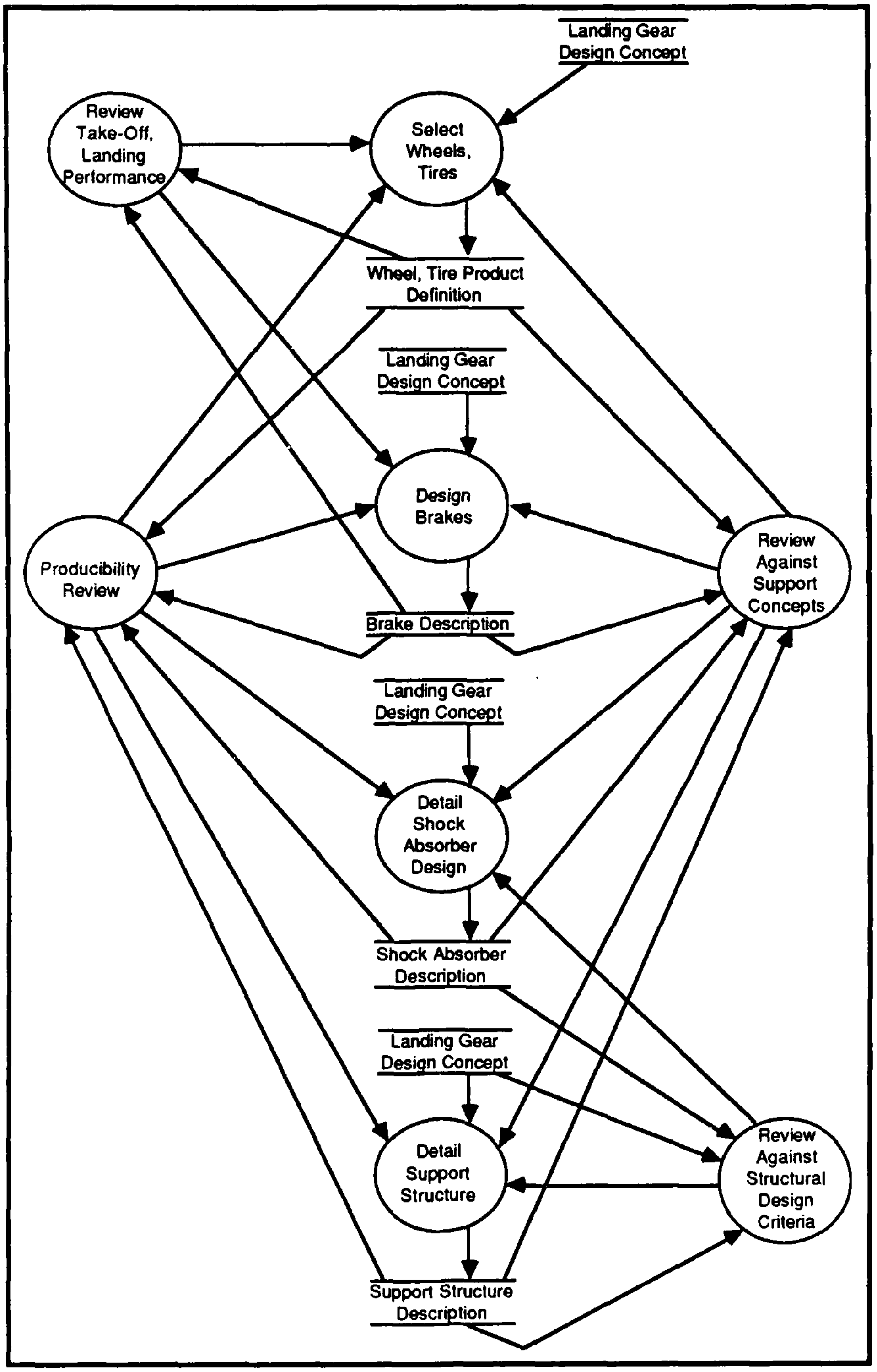


Figure C-9. Post Contractual Design Data Flow

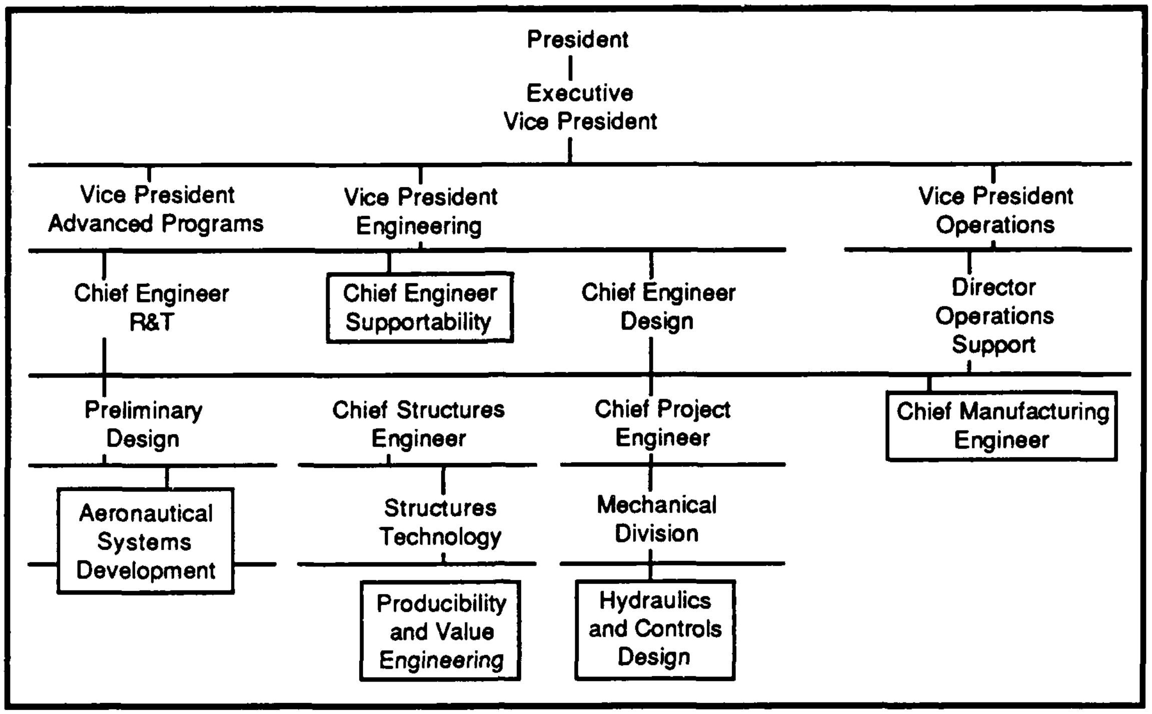


Figure C-10. Organizational Structure

CALS, ULCE, RAMCAD, and technology applications; and the operation of a field data record service.

(2) Producibility

During all phases of the product life cycle, producibility analyses and trade studies are performed by the Producibility and Value Engineering department. The line of authority for producibility decisions is through the Structures Technology Division, Chief Structures Engineer, and Chief Research and Technology Engineer, who reports to the Vice President--Engineering, as noted above. The producibility design process is shown in Figure B-7.

The office of the Chief Manufacturing Engineer (CME) performs manufacturing planning and engineering tasks that relate closely to producibility and support considerations. Functions and responsibilities of the CME include: the planning, development, and design of material handling equipment; the design, manufacture, and control of tools; and the planning and manufacturing of support equipment.

b. The Design Team

The individual organizations maintain technical expertise in their respective disciplines and provide engineering support for ongoing production contracts, such as the C-5B, C-130H, and L-100 projects; for advanced programs; and for research and technology contracts. When an aircraft program is identified as an important business opportunity for Lockheed, a project group is formed from engineering personnel who are assigned to the program on a relatively long-term basis. In the early phases of the design, the landing gear design team--including producibility and supportability engineers--may be called on to influence the requirements in the request for proposal (RFP). The design synthesis process takes shape by the combination of the design team's efforts.

Basic concepts of the landing gear are developed by the preliminary design organization and the landing gear designers. Project design resumes with detail development of the design concept. Draftsmen (2-5 years experience), checkers and supervisors, materials engineers, stress engineers, reliability and maintainability engineers, and a project engineer would be a typical example of a project design team. Producibility and value engineers basically intervene during project design to provide training, supervision, and continuity of the design. In conjunction to the project design team are the manufacturing process and planning engineers who help finalize the design. Through years of experience each member contributes his learned expertise to the design process.

Thus, although the organizational structure at Lockheed is not drawn strictly along matrix lines, the organization functions effectively as a matrix for major programs by providing personnel from the functional groups of the organization to project design teams.

5. Conclusions

Several conclusions can be drawn concerning how the current design process works:

- *Decisions are driven by the design definition process.*
- *The procedural architecture of the current process (Figure C-1) contains three levels of design decision-making*--individual, group, and technical management. The design decision is first attempted at the individual level, then at the group level, then, if all else fails, management enters the situation. The individual decision-making process is intertwined with the process of developing design definition. This is a result of the rapid build-up of design definition which is demanded to meet development schedules using current design technology.

- *The nature of the current design description is fragmentary, as illustrated by the data flow diagrams (Figures C-7 and C-8).*
- *Implementation concepts are formally documented; however, descriptions of intended functions (and unintended functions, if they have been identified) only are documented informally, if at all.*
- *Producibility and supportability considerations enter into the design process relatively late in the sequence of decisions.* This is a consequence of the fact that many important producibility and supportability decisions require the development of considerable design definition before the issues can be clearly identified. Production planning, logistics support analysis, the development of maintenance concepts and reliability and maintainability allocations, and goals occur outside the current design process and impact it through the requirements definition and design review procedures.

The organization of a project group to accomplish the development of design definition reflects these realities. The project group organization is structured along the lines of the Figure C-1 procedure flow. That is, the technical personnel assigned to the organization change over time as the sequential tasks of the current process require different landing gear design, analysis, and evaluation expertise.

D. DESIGN GOALS, CRITERIA, REQUIREMENTS, AND TRADEOFFS

During the design process, designers and engineers follow set goals, criteria, and requirements (both internal and those that are imposed by the procuring agency) in conjunction with tradeoff studies to search for near optimum solutions. This section details this process as it currently exists for landing gear at Lockheed. The objective of this chapter is to illustrate both the sizable number and the diversity of the requirements that must be considered in the design of a mechanical subsystem of an aircraft. The beginning sections consider only performance requirements. When producibility and supportability considerations are added, the resulting number of requirements becomes staggering.

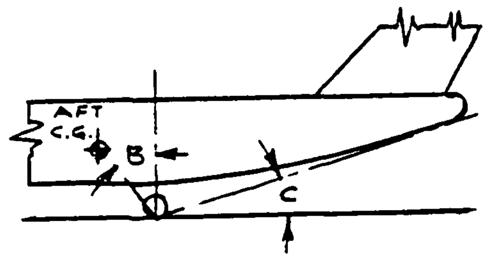
1. Requirements

A detailed list of design requirements for landing gears appears below. It is from Chapter 2 of Currey [Ref 4] and contains the generalized requirements for landing gear design from a designer's point of view.

It is helpful for designers to keep in mind that if requirements other than those imposed by the procuring agency are incorporated into the design, the resulting design is more universal in scope. If the initial design requirements and criteria are not compromised, then the additional requirements should be incorporated into the system. An example of differing requirements is the fact that some agencies require the left and right main landing units to be interchangeable, while others do not. This type of requirement (commonality/interchangeability) should be implemented whenever feasible for obvious economic reasons. Such implementation might lead to increased sales with other procuring agencies without a change in or a modification of the original design. The following requirements would provide a landing gear system that should be acceptable to both American and British authorities.

- Design the mechanism, doors, and support structure to permit lowering the gear at 1.6 of the calibrated stalling speed, with flaps retracted and at maximum landing weight.
- Unless there are other in-flight acceleration devices, design the gear and doors to withstand loads with gear down at 0.67 of the design cruise speed.
- The plane of each wheel should be vertical at design gross weight.
- The turnover angle should not exceed 63 degrees for land-based and 54 degrees for carrier-based aircraft .

- A tail bumper or skid should be provided.
- The tail bumper should not touch the ground when the main wheel is at the static position and when the aircraft angle of attack is appropriate to 90 percent maximum wing lift.
- Angle B shall not be less than Angle C, and Angle B shall not be less than 15 degrees.



- Shock strut normal oil level above the orifice should be at least 125 percent of piston diameter or 5 inches, whichever is less; otherwise test to demonstrate satisfactory shock absorption with performance impaired by foaming and/or leaking oil.
- The distance between the outer end of the shock-strut bearings should be at least 2.75 times the piston diameter.
- Shock-absorber units should be interchangeable left and right.
- Drop tests should be conducted to show that the shock absorber can absorb energy due to landing at 1.2 times the specified sink speed.
- Nosewheel-tire pressure should be based on allowable dynamic loads. These loads are 1.40 times static allowable for Type III tires, and 1.35 times static allowable for Type VII.
- Main-gear tire size should allow for 25 percent growth in airplane gross weight.
- Main-gear tire load rating shall not be exceeded under equal loading at maximum gross weight and critical CG position.
- On a multiple-wheel gear, ensure that when any one tire or wheel fails, the remaining tires and wheels can withstand the overloads imposed at maximum gross weight taxi.
- Wheel bead seat temperatures from braking should not exceed 350 degrees during normal and overload energy stops.
- Install fuse plugs to release tire pressure at, or less than, 400 deg. F tire bead seat temperature.
- Use forged aluminum-alloy wheels.

- Kinetic Energy absorbed by brakes = $0.0444\ WV^2/N$--where W is the weight of the aircraft, V is the power-off stall speed, and N is number of wheels.
- Normal brake energy is based on the greater of 1.15 times the recommended brake application speed, 1.0 times normal touchdown speed, or 1.1 times stalling speed in landing configuration.
- Locate brake lines on rear side of the gear structure to prevent damage.
- Provide emergency brake system, independent of the normal system, and capable of stopping the aircraft in the same distance as the normal system.
- Consider brake temperature in calculating brake energy capability.
- Install a parking brake capable of preventing roll on a 1-in-10 gradient or on a level runway with maximum takeoff power applied on one engine.
- Base rejected takeoff brake capability on a high-altitude, hot-day operation, with an initial temperature consistent with anticipated grounded time between landing and takeoff. In addition, assume a brake nearing the end of its recommended life.
- Antiskid systems shall be as reliable as the rest of the braking system, and cockpit warning lights shall indicate system failure.
- Uplocks shall be independent of door locks.
- Uplocks shall be releasable in an emergency by positive mechanical means.
- Downlocks should not be stressed by ground loads.
- Electrically operated locks must not be unlocked by electrical failures.
- Ground locks shall be provided, and their installation shall be foolproof.
- Retraction systems shall not use cables or pulleys, except in an emergency.
- An emergency extension system which is independent of the primary system shall be provided. The latter is defined as all parts stressed by ground loads.
- Do not use an emergency system requiring handpumping or cranking by the pilot.
- Minimize the use of sequencing mechanisms.
- In retracting mechanisms, do not use telescoping rods, slotted links, or cables.
- The maximum retraction time shall be 10 seconds. (Navy)
- The maximum extension time shall be 15 seconds. (Navy)
- Eliminate the possibility of mud or other material being trapped in cavities.

- Route all service lines, and locate all mechanisms and equipment, such that they will not be damaged by dirt, mud, water, or other material thrown by rotating wheels.
- MIL-L-878139 suggests a new requirement that the loss of any landing gear fairing door shall not result in the loss of the actuation power system command only, i.e., wires and hydraulic lines should not be routed on the doors.
- Close the doors after gear extension, and/or provide covers for burst tires or loose tread.
- Ensure that fuel tanks, lines carrying flammable fluids, and other hazard-creating items cannot be critically damaged by failure of landing gear parts.
- Stop the wheels from spinning in the retracted position.
- Provide enough steering power to steer the aircraft without the necessity of forward motion.
- Provide an emergency system capable of steering the aircraft without interruption if the normal steering system fails.

The list below is a summary of military specifications and paraphrased titles that are commonly used during landing gear design. Each of these specifications may in themselves contain hundreds of hard requirements, and often more than one military specification may apply per design step.

PRIMARY MIL SPECIFICATION SUMMARY (Titles Paraphrased)

MIL-A-8629	Drop Tests (see also MIL-T-6053)
MIL-A-8860	Airplane Strength--General Specifications
MIL-A-8862	Ground Handling Loads
MIL-A-8865	Airplane Strength & Miscellaneous Loads
MIL-A-8866	Strength & Rigidity Reliability
MIL-A-8868	Airplane Strength Data & Reports
MIL-B-8075	Anti-Skid
MIL-B-8584	Brakes--Control Systems
MIL-C-5041	Tire Casings
MIL-D-9056	Drag Chute
MIL-H-5440	Hydraulic Components
MIL-H-5606	Hydraulic Fluid

MIL-H-8775	Hydraulic System Components
MIL-L-8552	Shock Absorbers--AFSC and USN
MIL-P-5514	Packings--Shock Struts; also 0-Rings and Glands
MIL-P-5516	Packings--Shock Struts
MIL-P-5518	Pneumatic Components
MIL-P-8585	Primer--Wheel Wells
MIL-S-8812	Steering Systems
MIL-T-5041	Tires
MIL-T-6053	Drop Tests (also MIL-A-8629)
MIL-T-83136	Tie Down Requirements
MIL-W-5013	Brakes and Wheels
MIL-STD-203	Controls and Displays in Flight Station
MIL-STD-805	Tow Fittings
MIL-STD-809	Jacking Fittings
MIL-STD-878	Tires and Rims Dimensioning and Clearances
MIL-STD-568	Corrosion Prevention and Control

2. Detailed Design Considerations

After the requirements have been outlined in the concept formulation phase of the preliminary design, a layout is prepared to locate the main and nose landing gears longitudinally and laterally on the aircraft. A summary of the steps performed during the layout phase of the design of the landing gear is detailed as follows from Reference 4.

(1) Locate mean aerodynamic chord on side view.

(2) Locate forward and aft CG limits on the MAC.

(3) Locate vertical CG positions at these forward and aft limits.

(4) Locate main gear at the best position for structural pick-up.

(5) Is the main gear at about 50 to 55 percent of the MAC?

(6) Draw line at 15 degrees to vertical from the aft CG to the ground. Draw line from bottom of tail bumper at 12 to 15-degrees to the horizontal. Where this line meets the 15-degree line, place the centroid of the main landing gear wheel(s). Adjust main gear position accordingly.

(7) Locate nose gear.

(8) Calculate static and dynamic landing gear loads.

(9) Make preliminary tire selection.

(10) Draw in the tires at their correct loaded radii, and locate static axle center.

(11) Approximate the shock strut travel and show the Compressed and Extended shock strut positions on the drawing for both nose and main gear. Check the ground angles with the tires at these locations.

(12) Show the landing gear structure by stick diagram.

(13) Calculate Turnover Angle and adjust landing gear if necessary.

(14) Check pitch/roll clearances and all other conditions with various combinations of flat tires and flat struts.

More detailed considerations are necessary in the design process as the landing gear system begins to take shape. After the initial overall layout has been completed, and the specific requirements have been reviewed, the designer must transform his ideas, sketches, and layout drawings into practical hardware. Specific items that should be observed are:

Flotation	Towing
Stowage	Jacking
Structural Support & Load Paths	Ground Safety Locks
Braking	Uplocks/Downlocks
Steering & Turn Radius	Using Existing Components
Operational Characteristics	

At this stage in the design process, the following questions must be considered by the designer [Ref. 4]:

(1) Is flotation adequate?

(2) How does flotation compare with similar aircraft?

(3) Is it necessary to retract the gear?

(4) Are the kinematics simple and devoid of as many three-dimensional motions as possible?

(5) Has sequencing been minimized?

(6) If sequencing is used, has the most reliable method been chosen?

(7) Are wheel well clearances adequate?

(8) Has free-fall capability been considered?

(9) Have tracks and rollers been eliminated as much as possible?

(10) If a landing gear pod is used, has a minimum-drag configuration been developed?

(11) Are the load paths simple?

(12) Have redundant structures been avoided?

(13) Have the flight operations been considered to such an extent that brake cooling times can be checked, and are the brakes satisfactory when operating at the temperatures resulting from these cooling times?

(14) Is the turning radius satisfactory?

(15) What is the nose landing gear steering angle?

(16) Have the steering disconnects been considered?

(17) Have rudder pedal steering and handwheel steering been analyzed with respect to associated rudder travel and turns on the handwheel?

(18) Has shimmy damping been considered?

(19) Will the wheels be stopped during retraction, and if so, how?

(20) What is the speed after takeoff at which the gear will be retracted, and what retraction time is allowed?

(21) What is the speed at which the gear must be capable of being lowered, and what extension time is allowed?

(22) Have jacking provisions been provided, and is the jacking ball capable of taking severe abuse?

(23) Have towing provisions been provided, along with devices to ensure that towing forces will not damage the gear?

(24) Have ground safety locks been provided that are capable of withstanding the retraction loads?

(25) Has consideration been given to the provision of uplocks and downlocks, and in particular, has an absolutely foolproof method been devised to ensure that means will always be available to release the uplock?

(26) Has consideration been given to the use of existing components to eliminate non-recurring costs, and to use parts of proven reliability?

3. Tradeoffs

The items listed here only represent some of the design issues that are defined during parametric and tradeoff studies of landing gear systems.

- Number and size of tires vs. cost, weight, and flotation
- Drag vs. weight and cost
- Location of main gear (wing, nacelle, or fuselage) vs. cost, weight, and performance
- Shock strut travel vs. load factor vs. weight and cost
- Brake material selection
- Use of auxiliary braking systems
- Crosswind landing gear system vs. weight, cost, reliability, maintainability
- Electric vs. hydraulic systems for retraction, extension, and brakes.

Additional design considerations, requirements, and tradeoffs for shock absorbers, tires, brakes/wheels/skid control, kinematics, and steering systems are discussed in Reference 4.

4. Producibility

Producibility and value engineers follow strict guidelines to ensure that the design follows set requirements for manufacturing. Two different examples are presented to show the varying guidelines that must be adhered to during the design process. The first example details the checkoff list that would be followed for forged items. The second example examines the installation of assemblies and how the parts fit together.

Forging Example:

Below is a list of the requirements that are used during the design of an assembly that requires forging. This is part of a checkoff list that is used during the examination of a detailed drawing for a specific part of the landing gear that must be manufactured by forging.

GENERAL:

(1) What is it and its function?

(2) Is it necessary?

(3) Why is it the way it is?

(4) Is there a simpler design?

(5) Can it be combined with another part?

(6) Has the effect of quantities on the type of part been considered?

(7) Are standard fabrication and assembly methods used?

(8) Can the number of operations be reduced?

(9) Can the same part be used for the left and right hand?

(10) Does the drawing give sufficient information for fabrication?

(11) Has the most economical manufacturing process been utilized? (sheet metal, extrusion, forged, cast, machined, welded, etc.)

(12) Can tolerances be loosened?

(13) Can standard parts in current inventory and approved status be used?

ELIMINATING AND COMBINING PARTS:

(1) Can an attaching flange be combined on an existing part to eliminate clips and angles?

(2) Can the part be stiffened with integral beads rather than stiffeners?

(3) Can chem-milled part incorporate doublers?

(4) Can an extrusion incorporate webs, angles, tees, etc., into a single part?

(5) Can several details be economically incorporated into casting or forging?

(6) Can casting or forging be made more economically by making the left- and right-hand parts from a single casting or forging?

(7) Can a net precision forging eliminate built-up parts or parts machined from forgings?

(8) Can left- and right-hand parts be combined by adding a lug, a tab, or another redundant feature?

(9) By superimposing left- and right-hand hole patterns and permitting open holes, can the part be no handed?

(10) Check once more to see if the opposite part can be eliminated.

PROCESSING:

(1) Is special processing required?

(2) Are process specifications in existence for all required operations?

(3) If the process is new, does the drawing give sufficient information?

(4) Does the part exceed processing equipment capacity?

(5) Is the process compatible with design requirements?

(6) Does the processing affect the strength of the part?

(7) Does the process embrittle the part?

(8) Does the process require post processing operations?

(9) What are the inspection requirements? What tests are required?

(10) Are dissimilar metals isolated?

(11) Does the process adversely affect the sequence of assembly?

(12) Does the process affect the corrosion resistance of the material?

(13) Does the processing result in intolerable residual stresses?

(14) Can a more corrosive resistant material be used in the design?

(15) Are finish callouts compatible with material callouts?

(16) Are before and after plating dimensions specified when required?

(17) Are finish callouts compatible with finish specification requirements?

(18) Were environmental conditions such as high temperature, particularly corrosive environment or abrasive conditions, considered in the finish selected?

(19) Has the model finish specification been consulted for peculiar finish requirements?

MACHINING:

(1) Does the part use standard machine operations?

(2) Are the most economical types of machine operations used?

(3) Can operations be combined? Or eliminated?

(4) Can the material be easily machined in final heat treat condition?

(5) Is heat treatment sequence critical to machine operations?

(6) Can operations be made in one setup?

(7) Is baking or shot peening required?

(8) Can the tolerance on any feature be relaxed?

(9) Is the hole pattern correctly dimensioned for tool make and interchangeability?

(10) Can blind, flat-bottomed holes be eliminated?

(11) Is the class of cylindrical fit as loose as possible?

(12) Is the concentricity requirement to the D.II. standards?

(13) Can Hydrotel operations be eliminated?

(14) Are threads, serrations, and involute splines standard?

(15) Does the drawing show thread relief or imperfect thread run out?

(16) Is plunge cut optional?

(17) Is surface finish compatible with function?

(18) Will warpage occur during machining?

(19) Are springing allowances required?

(20) Is additional material for clamping shown?

(21) Are radii shown as min/max standard radii?

(22) Have blended radii been eliminated?

(23) Is cutter mismatch shown and generous tolerance on cutter run out?

(24) Are standard cutters used for slots?

(25) Is material allowance sufficient to make the part considering material tolerances, hold down provisions, chucking, etc.?

(26) Is extra material required to be machined off in order that the finished part will pass magnetic particle inspection?

(27) Will permanent mold casting save machining?

(28) If the part has a hole pattern, is it made deliberately and obviously non-symmetrical so as to avoid improper installation?

(29) Is part dimensioned suitable for numerical control programming?

(30) Is depth of milling and cutter diameter within machining limitations?

(31) Is a casting, extrusion, or forging a more economical material form?

FORGINGS:

(1) Is part number location correct?

(2) Is parting line location most economical?

(3) Are tooling pads required?

(4) Is material call out satisfactory?

(5) Are draft angles adequate?

(6) Can coining be profitably used to eliminate machining?

(7) Heat treatment required?

(8) Should critical grain direction be specified?

(9) Are internal and external radii adequate

(10) Is material available to fill in forging?

(11) Are rib and cavity proportions okay?

(12) Is machining properly located to rough forgings, first cut shown, tooling points indicated, identification pads indicated?

(13) Are tolerances adequate?

(14) Can contoured surfaces be eliminated?

(15) Is flatness tolerance specified?

(16) Can shape of part be modified to simplify dies?

(17) Are double forgings economical?

(18) Will web thicknesses allow proper forgings?

(19) Can machined part be made from extreme limit forging? Are bosses elongated?

(20) Have mismatch allowances been checked?

(21) Forging designed to use standard machine tools?

(22) Has a low draft press forging been considered?

(23) Is the total number of parts sufficient to justify forging?

(24) Have allowances been made for possible salvage of rejected parts?

(25) Have NET forgings been considered? Are they economical?

(26) Have reference lines, points, or other means been incorporated into forging to permit raw material inspection and minimize shop errors in setup and machining?

(27) Can flat back forging be used?

(28) Has adequate die closure tolerance been allowed?

(29) Can material removal be symmetrical?

(30) Are dimensions to mold lines and easily determined features, rather than imaginary construction lines, on the forging?

(31) Are angles and contour controlled by coordinate dimensions originating from die impression on the same side of tool as contour or angle?

PART IDENTIFICATION:

(1) Is marking as permanent as the part being marked?

(2) Has impression stamping been allowed where possible?

(3) Is material compatible with impression stamping?

(4) Does drawing show location of permanent marking?

(5) Does drawing call out identification specification?

STANDARDS:

(1) Has the use of design standards been considered?

(2) Are inspection requirements on the part adequate for its application, e.g., magnetic inspection, etc.

(3) Does design use the minimum number of types of standards possible?

(4) Is the finish of the standard part adequate for its environment?

Installation Example:

Below are the details of requirements for the installation of assemblies. The producibility engineer must ascertain that all related assemblies fit together and function in accordance with the design objective. Many items on the checkoff list are similar to those in the forging example.

INSTALLATION EXAMPLE:

GENERAL:

(1) What is it and what is its function?

(2) Is it necessary?

(3) Why is it the way it is?

(4) Is there a simpler design?

(5) Can it be combined with another part?

(6) Has the effect of quantities on the type of part been considered?

(7) Are standard fabrication and assembly methods used?

(8) Can the number of operations be reduced?

(9) Can the same part be used for the left and right hand?

(10) Does the drawing give sufficient information for fabrication?

(11) Has the most economical manufacturing process been utilized? (sheet metal, extrusion, forged, cast, machined, welded, etc.)

(12) Can tolerance be loosened?

(13) Can standard parts in current inventory, with approved status, be used?

ELIMINATING AND COMBINING PARTS:

(1) Can an attaching flange be combined on existing part to eliminate clips and angles?

(2) Can several details be economically incorporated into casting or forging?

(3) Can casting or forging be made more economically by making the left- and right-hand parts from a single casting or forging?

(4) Can left- and right-hand parts be combined by adding a lug, a tab, or another redundant feature?

(5) Check once more to see if opposite part can be eliminated.

ASSEMBLY:

Major Components:

(1) Is the production breakdown most efficient?

For contract quantity and facilities
For the production rate
For manpower distribution
For internal stuffing

(2) Are component sizes within handling, shipping, and processing limitations?

(3) Are lift points designated for major assemblies?

Joints:

(1) Are joint attachments and fittings held to a minimum and are they readily accessible?

(2) Will the joints meet interchangeability requirements?

(3) Are mating tolerances restrictive?

(4) Will relative motion in joints remove the protective coating?

SUBASSEMBLIES:

(1) Do assemblies incorporate the free body unit design principle?

(2) Is the assembly sequence satisfactory?

(3) Are detail tolerances compatible with assembly requirements?

(4) Is take up adjustment necessary or required?

(5) Is sufficient access provided for assembly operations? For standard tools and equipment?

(6) Does the assembly permit access for service inspection and for maintenance of functional equipment?

(7) Are the interchangeability requirements controlled dimensionally or by tooling?

(8) Are dissimilar metals properly insulated?

(9) Will relative motion (movable parts) remove all protective finish?

Fasteners:

(1) Are standard fasteners used? Can the number be reduced?

(2) Has the most economical fastening method been selected?

(3) Is minimum fastener edge distance provided on all parts?

(4) Is additional fastener edge distance provided on parts requiring fitting on installation?

(6) Is standard tool clearance provided?

(7) Is clearance sufficient for wrenching?

(8) Is selection of hole sizes compatible with function, assembly, and interchangeability requirements?

(9) Is fastener head direction left to manufacturing option when possible?

(10) Are adequate fastener instructions called out?

(11) Is bolt torque specified if nonstandard?

(13) Are washers provided?

(14) Are safetying requirements for fasteners met?

(15) Are self-locking nuts used wherever possible?

(16) Are dissimilar metals properly insulated or avoided where possible?

(24) Does fastener have proper chemical finish?

STANDARDS:

(1) Can standard parts which are in current inventory, with approved status, be used?

(2) Has the Standards Group been consulted if the desired standard was unavailable in the manual?

(3) Has the order of precedence for use of standards per MIL-STD-143 been considered?

(4) Has reliability of standard part been considered and, if so, is it adequate for the end item of which it is a component?

(5) Have the use of design standards been considered?

(6) Does design use the minimum number of types of standards possible?

(7) Is the finish of the standard part adequate for its environment?

5. Supportability

Quantitative and qualitative supportability requirements imposed by the customer and/or by the LSA process are issued as Supportability Design-to Requirements (SDTRs) to the designers. A partial list of requirements, as developed by Lockheed's supportability group, is given as follows.

- To minimize malfunctions, there should be no exposed close tolerance interfacing components.
- To minimize environmental exposure and task complexity, component removal and replacement should not expose bearings and/or other internal parts.
- To minimize preventive maintenance and increase durability, critical component surfaces should not be exposed to the environment when the gear is extended.
- To increase combat mobility, high frequency maintenance should not require unique support equipment (SE).
- To reduce spares support, design should be simple with few parts and durable to the operating environment.
- To increase functional reliability by increased redundancy that minimizes the effects of single critical component failure or battle damage; to reduce SE requirements for component replacements by using individual gear operation as a self jacking feature; and to provide fly out capabilities from a hostile or NBC contaminated operational environment, each main landing gear strut should have independent operational capability in the air and on the ground.
- To minimize manhours, skill levels, support equipment, and exposure of greased bearings to adverse (sand, dust, and mud) physical environments, the wheel design should allow tire replacement without disturbing hub components.

- To reduce spares support burden, the retraction system needs to be simplified to reduce the number of different critical functional parts.
- To enhance durability, increase reliability, and reduce preventive maintenance, the retraction system needs to eliminate exposed loose tolerance functional components (such as guide shoes, tracks, and ballnuts) that are susceptible to environmental- (sand, dust and mud) and rigging-induced malfunctions.
- To reduce need for alignment support equipment; to reduce tire susceptibility to damage from brake heat; and to facilitate rapid tire replacements, the wheel/tire assembly should not require direct engagement with the brake assembly.
- To minimize the number of mounting fasteners, tire-to-brake torque transfer should continue by passive means (heavy splines).
- To minimize skill levels, support equipment, and the risk of induced damage, mounting of wheel/tire assembly should not require close tolerance alignment
- To preclude the needed spares purchasing or special storage requirements, spare wheel/tire assemblies should not include special care items (bearings).

There is an increasing emphasis in aircraft design on getting supportability requirements into the design process as early as possible. Designers and engineers should be aware of the impact that the components they design will have on the overall supportability of the design. Consideration of alternative or possible uses of a designed object should be included during conceptual, preliminary, and production design. Knowledge of past experiences only allow designers or engineers the ability to compare or parallel current design strategies. Creativity must be present for the expansion or growth of technology in current designs to afford new and changing design methodology for future projects.

6. Analytical Tools

Much of the performance analysis of landing gears is straightforward and is performed by hand using well-tested engineering methods. This is especially true in the aircraft conceptual design and in the early stages of the design of the landing gear itself. Perhaps the most sophisticated analysis is applied to determine the dynamic landing and braking loads. This analysis involves finite element modeling (Nastran) of the aircraft primary structure, modeling of low speed unsteady aerodynamics, and time-domain response of the landing gear itself, which acts as a nonlinear spring.

Recent developments in analytical software have been used in aircraft landing gear to help:

- Predict landing gear/soil interaction
- Establish roughness criteria for LC-130 Antartic operations
- Determine C-130 response to bomb-damage/repaired runways
- Develop operations-on-soil prediction techniques.

In the prediction of landing gear/soil interaction, theoretical soil models are coupled with simulations of aircraft landing impact, runout, turning, and takeoff. Roughness criteria for LC-130 Antartic operations for the Naval Air Systems Command was established to evaluate the operational limits for the C-130 in the Antarctic.

A program to predict the C-130's response to bomb-damage/repaired runways was conducted for the Air Force Engineering and Services Laboratory (AFESC) with the desired objective to accurately calculate C-130 aircraft landing gear and structural loads that occur from taxiing, taking off, and landing on bomb-damaged/repaired airfields. Operations-on-soil prediction techniques were initiated to improve the existing computerized techniques for predicting tactical aircraft operations on soil surfaces.

There are no integrated analytical tools presently used to process supportability requirements, to identify goals, to allocate goals, or to conduct trade-offs. Currently, these functions are done using isolated models; for example, those listed in Figure D-1.

Numerical design optimization techniques are not currently used for landing gear design for performance or producibility at Lockheed Aeronautical Systems Company. Instead, a manual or interactive optimization process is normally used which involves consideration of a large number of alternatives and elimination of non-optimal candidates by successive application of design criteria. The criteria are applied in stages, and only the minimal amount of design definition is developed at each stage.

An example of the logic applied to select the best alternative design concept for supportability factors is presented in Figure D-2. The tradeoff represented is between the mean time between critical events (MTBCE) and the mean time to repair (MTTR). Weighting factors are not currently applied to any terms appearing in the supportability measure, nor is supportability weighted relative to producibility, performance, cost, or schedule.

AFEM - Airlift Fleet Evaluation Model, computes fleet size, sortie generation rate, and utilization rate for strategic airlift mission

AFSEM - Airlift Fleet Supportability Evaluation Model, similar to AFEM, but has explicit modeling of spares delays. NMC rates are output.

CARSRA - Computer Aided Redundant System Reliability Analysis, Markov process model for dynamically redundant systems with transients and coverage.

GASPIN - General Aircraft Sizing Program Insert, a subroutine of GASP, consisting of a series of regression models, to estimate R, M, and LCC for conceptual design transports.

GOAL - General Optimal Allocations Program, allocates R and M requirements to lower level work unit codes according to improvability allowances.

Manpower model, yields the manpower needs per unit and the skills need of the maintenance organizations.

PAYBACK - Payback Period and LCC Computation, performs breakdown analysis and life cycle cost with option to include cost of abort, cost of downtime.

RACS - Reliability of Aircraft Control Surfaces, models self-repairing integrated flight/propulsion control effectors.

RAMSCES - Reliability, Availability, Maintainability and Support Cost Estimating System, 99 R and M, LCC parameter values for historical, baseline, predictive, or allocations configurations.

SASM - Supportability Assessment Simulation Model, simulates maintenance, including servicing, battle damage repair, weapons reconfiguration for a single base squadron.

SURE - Semi-Markov Unreliability Range Evaluator, NASA-Langley program currently in Beta test at Lockheed-Georgia, models reconfigurable computers to find bounds on system reliability.

TARA - Tactical Resource Analysis Model, simulates multi-base airlift with spares delay, but in less detail than VISA.

VISA - Vehcle Integrated Supportability Assessment, much like SASM, but models multi-base tactical missions with deferred maintenance and on-board spares.

WSR - Weapon System Reliability model, computes air vehicle level reliability with all redundancy effects included.

Figure D-1. Supportability Analysis Tools

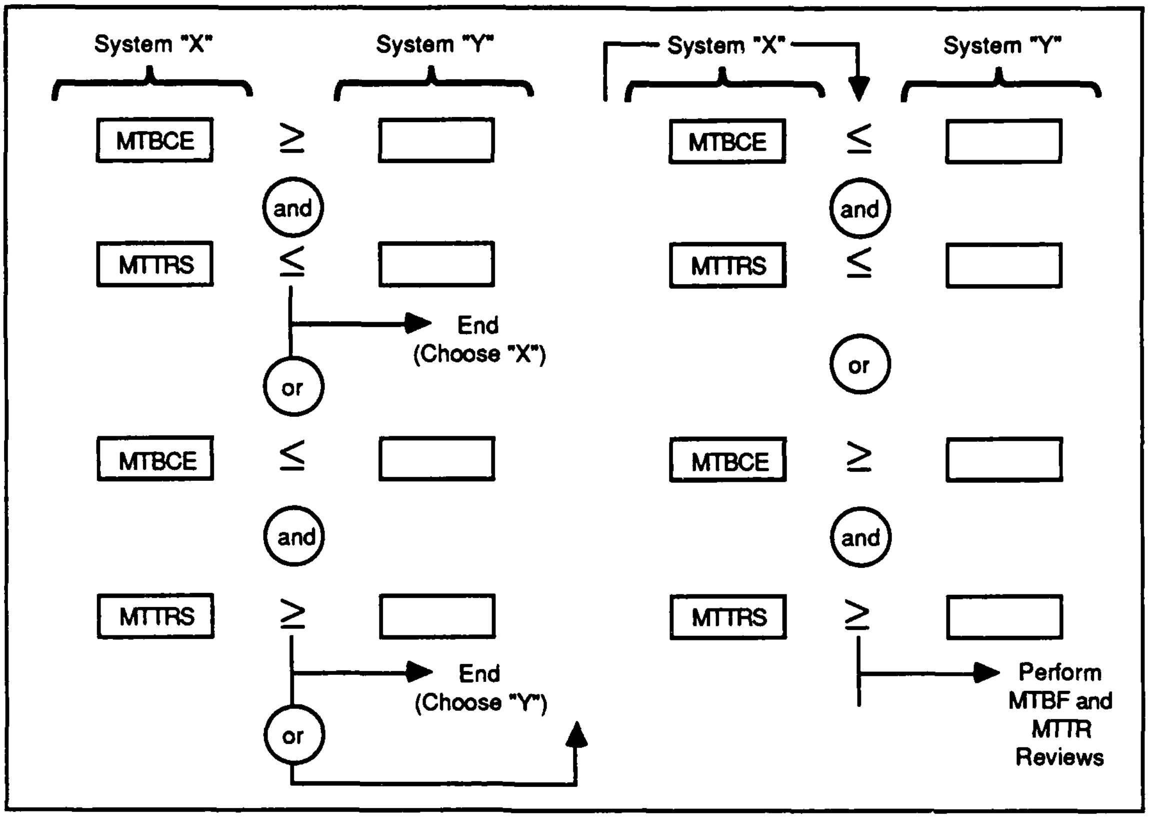


Figure D-2. Comparative Evaluations

7. Decision Process for Hierarchical Assignment of Goals and Requirements

Hardware design for performance follows a relatively fixed assignment of function and associated goals and requirements to system elements. A similar point of view characterizes the current approach to producibility.

The measure of supportability goals for an airlifter system and its elements is illustrated in Figure D-3. The interval of critical events (mean time between critical failure divided by the mean time between critical failure event, MTBCF/MTBCFE) and the downtime of critical events (mean time to repair, MTTR) are the key drivers that influence the operational readiness measures of Sortie Generation Rate (SGR), Mission Capable Rate (MCR), Operational Availability (A0), and Readiness (R). Further, the MTBCF value directly determines the mission reliability--flight (D)--while the MTBCFE determines the mission reliability--ground operations (C). The hierarchy of supportability (S̲) function is noted in the right-hand column of Figure D-3:

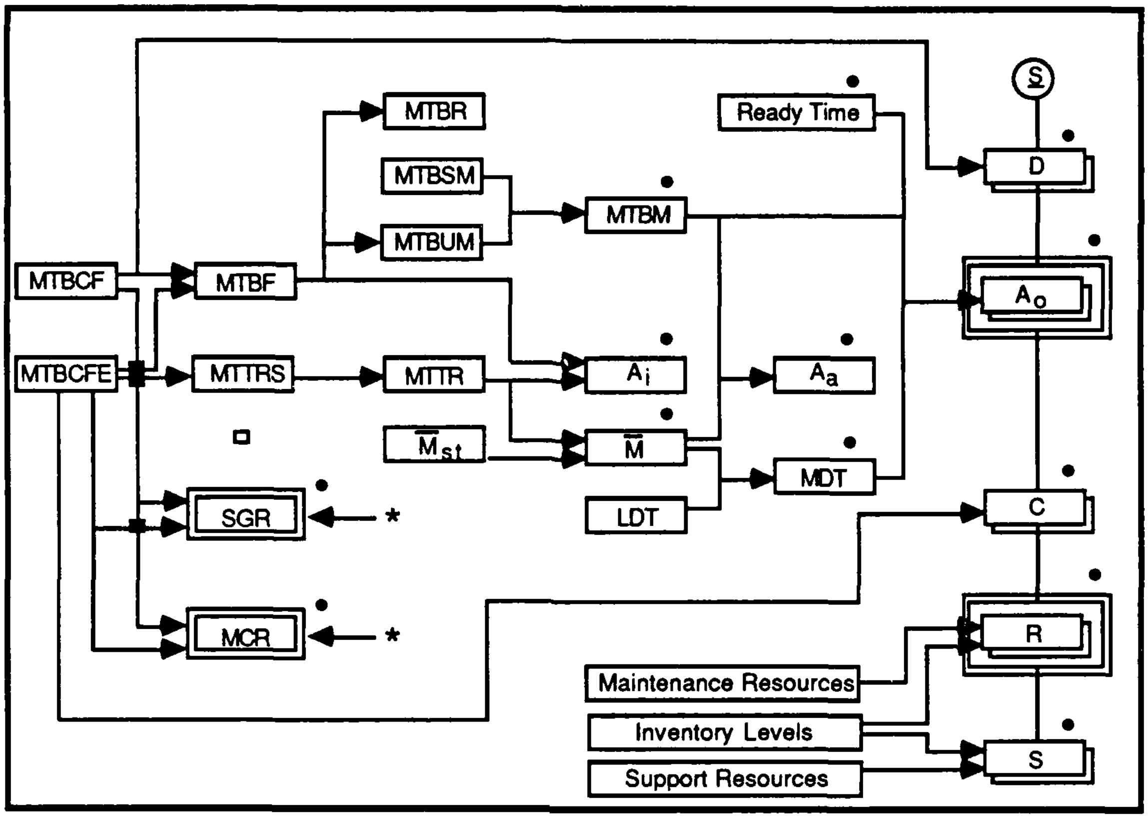


Figure D-3. **Hierarchy of Tradeoff Elements**

- D: Dependability
- A: (Operational) Availability
- C: Capability
- R: Readiness
- S: Sustainability.

Expressions exist for each of these factors as reported in Ref. 5.

8. Arbitration Among Conflicting Design Goals

Conflicts among design goals are resolved through a process of examining the impacts of design alternatives or technical and cost figures of merit, arriving at a documented management decision regarding the preferred alternative.

The formal trade study process is summarized in Figure D-4. This process is initiated when potential alternatives to a baseline configuration are identified. If these potential alternatives require a detailed trade study analysis, a schedule is established and

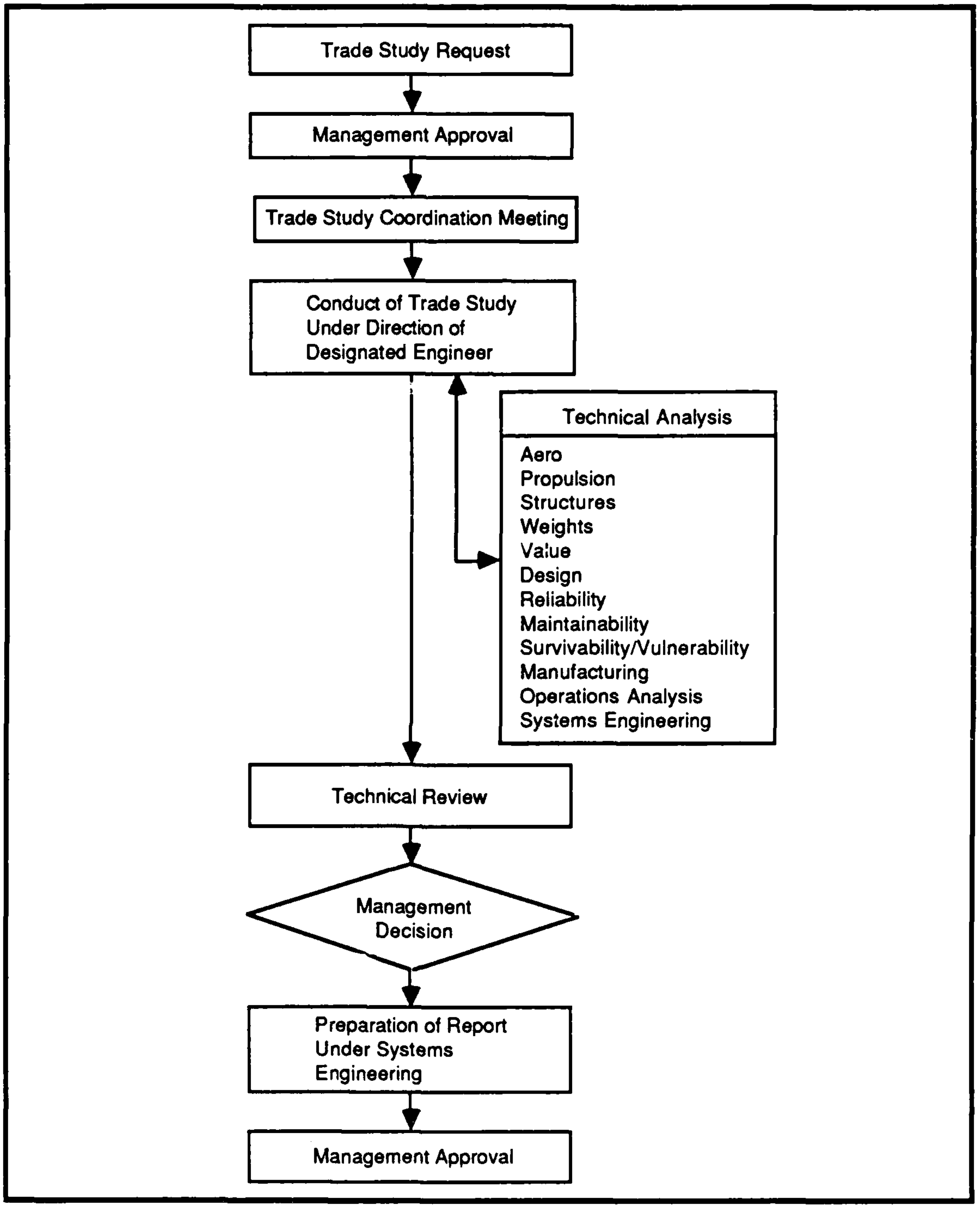


Figure D-4. Trade Study Process

design activity is initiated to provide the necessary design details of the baseline and each alternative. Based on these design activities, incremental acquisition, logistic support, and operating costs are estimated, along with the effect on operational utility, maintainability, reliability, and availability for both the direct effects associated with each alternative and the indirect effects that arise from the need to resize the aircraft to assure that all performance requirements are still met. A bottom-line life cycle cost impact is calculated, a final check to ensure that the alternatives satisfy applicable requirements is made, and an assessment of difficult-to-quantify parameters, such as operational utility is made. Management decisions are based on the summary impact information. The trade study is then documented in a prescribed format, stored in a trades data base, and routed to specialists and managers for final evaluation, decision, and approval.

9. Conclusions and Observations

A significant number of design requirements for landing gear are of a qualitative nature. Thus, any ULCE design process for landing gear must be very good at evaluating a large number of qualitative design possibilities to arrive at a good overall design. Qualitative factors appear to dominate in considerations of producibility and supportability, although quantitative measures can be developed for some aspects of these considerations. Procedures for handling mixtures of qualitative and quantitative considerations in trade studies will be a critical part of any ULCE design process.

E. ULCE ARCHITECTURE

1. Introduction

There are two separate aspects of a proposed ULCE architecture addressed in this section. These are:

- The concepts and structure of the ULCE design process.
- The application of new technology that will support the implementation of the new process.

The approach to the proposed architecture is based on shortcomings of the current design process and on opportunities afforded by new computing technology. There are a number of critical flaws in the current design process that are amenable to major improvements by the implementation of ULCE. Among the key objectives of the proposed architecture are listed below.

- Modify the design sequence to bring <u>all</u> design related activities and considerations into the design process at an earlier stage, thereby facilitating iterative cycles of design and a more complete tradeoff and optimization between the often times conflicting requirements of performance, supportability, producibility, and cost.
- Increase the visibility of design decisions.
- Improve the retention and accurate transfer of product specific knowledge or information developed and/or required throughout all stages of the design process.

a. Shortcomings of the Current Design Process

The current design process is data driven--design specifications are the deliverable items and the process is managed to produce these on schedule. Design decisions are buried in specification data. Specifications may contain built-in contradictions reflecting decisions that were never made. These contradictions may not show up until the product is manufactured, tested, or deployed.

The development schedule applies pressure to quickly develop deliverable specification data, and results in a depth-first search through alternative designs. Performing this search successfully requires a sequence of design procedures which produces a rapid build-up of design data. Attributes of the design, which are viewed as obstacles to generating required deliverables, are specified early in the sequence. A

sequential design process (Figure E-1) necessarily reduces the design freedom available to meet requirements as successive attributes are specified. Producibility and supportability analyses, which require a certain level of design specification to be performed, cannot be performed because of lack of information. Therefore they are put off until the tail end of the design process.

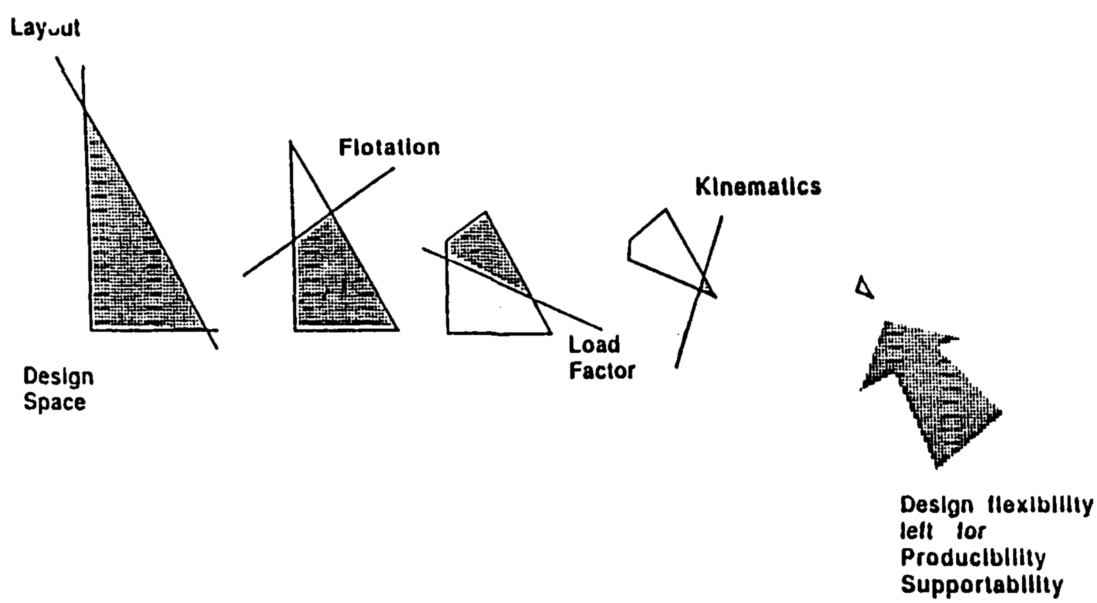


Figure E-1. "Freeze and Squeeze" in the Current Sequential Design Process

One solution to the problem is to require supportability and producibility analyses earlier in the design process. This approach will not solve the whole problem, however. Designs are balanced by decision making, not analysis. Specification of design practice should encourage technical management by decisions. The timetable for making these decisions will be program-specific and will depend on requirements flow, technology, decision support, and analysis tools. ULCE must address the need for a *Meta-Design Process* to design the design decision-making process. Emphasis on the design decision timetable and recognition of the role of engineering analysis in evaluating design alternatives will help to avoid problems in the current specifications for design practice. Figure E-2 illustrates such a problem: reliability and maintainability analyses are not called out in time to support decisions that must be made as part of the LSA process.

SUPPORTABILITY TECHNICAL TASKS FUNCTION/TASK REFERENCE NO.			PROGRAM PHASES				
			PC	C	DVL	FSD	P/D
LSA	--	Use Study Development (201)	x	x	x	x	x
LSA	--	Comparative Analysis (203)	x	x	x	x	
LSA	--	Design Reviews (103)		x	x	x	
LSA	--	Functional Requirements (301)		x	x	x	
LSA	--	Support System Design (302)		x	x	x	
LSA	--	Standardization Analysis (202)		x	x	x	
LSA	--	Technology Studies (204)		x	x		
LSA	--	Design Factors Definition (205)		x	x	x	x
LSA	--	Tradeoff Studies (303)		x	x	x	
LSA	--	Tests and Evaluations (501)		x	x	x	
LSAR	--	Resource Requirements Identification (401)				x	x
LSAR	--	Early Fielding Analyses (402)				x	x
LSAR	--	Support Analyses (403)					x
M	--	Vendor Technical Liaison (102)				x	x
M	--	Design Reviews (103)			x	x	
M	--	Change Control (104)				x	
M	--	Models Development (201)	ⓧ	ⓧ	ⓧ	x	
M	--	Allocations (202)		ⓧ	ⓧ	x	
M	--	Predictions (203)			ⓧ	x	
M	--	FMEA (204)	ⓧ	ⓧ	x	x	
M	--	Analyses (205		ⓧ	ⓧ	x	
M	--	Design Criteria (206)	ⓧ	ⓧ	ⓧ	x	
M	--	LSA Inputs (207)				x	x
M	--	Testing Analyses (301)				x	x
R	--	Vendor Technical Liaison (102)				x	x
R	--	Design Reviews (103)				x	x
R	--	Change Control (104/105)				x	x
R	--	Models Development (201)				x	x
R	--	Allocations (202)	ⓧ	ⓧ	x	x	x
R	--	Predictions (203)		ⓧ	ⓧ	x	x
R	--	FMECA (204)			ⓧ	x	x
R	--	Analyses/Parts Controls (205-208)				x	
R	--	Testing/Analyses (209-30x)				x	

Key: General Requirement Denoted by "x".
LSA Program Additional Need Indicated by ⓧ

Figure E-2. Supportability Technical Tasks

b. Design Information Loss

One of the significant shortcomings of the current design process is the loss of information and knowledge about the design itself between stages of the design process. This loss contributes not only to delays in the process but is a major contributor to the design errors that enter because of inconsistencies in the design representation or lack of knowledge about preceding design decisions. The problem of information loss in the

design process has been recognized for some time and is described in a number of articles on improving the design process [Refs. 6, 7].

An example of the typical information that might result in deleterious downstream changes to a design is described by Popplestone, et. al., wherein the details of a stress analysis by engineering are not conveyed in the parts drawing to manufacturing. This can be illustrated in Figure E-3.

The analysis of a load bearing support might show stress concentrations at the sharp junction of two faces of the support. A simple way to reduce the stress concentration is to add a rounded fillet at that point. However, the resulting engineering drawing shows only the dimensions of the part with no indication of the relationship between the radius of the fillet, the thickness of the support, and the resulting stresses. A manufacturing enginer might modify the radius to simply tooling requirements or reduce machining operations; the result, a potential reduction in the load bearing capability of the support. This is not an atypical situation and is a miniscule sample of the type of knowledge and information that is not transferred across the many such interfaces in the current design process.

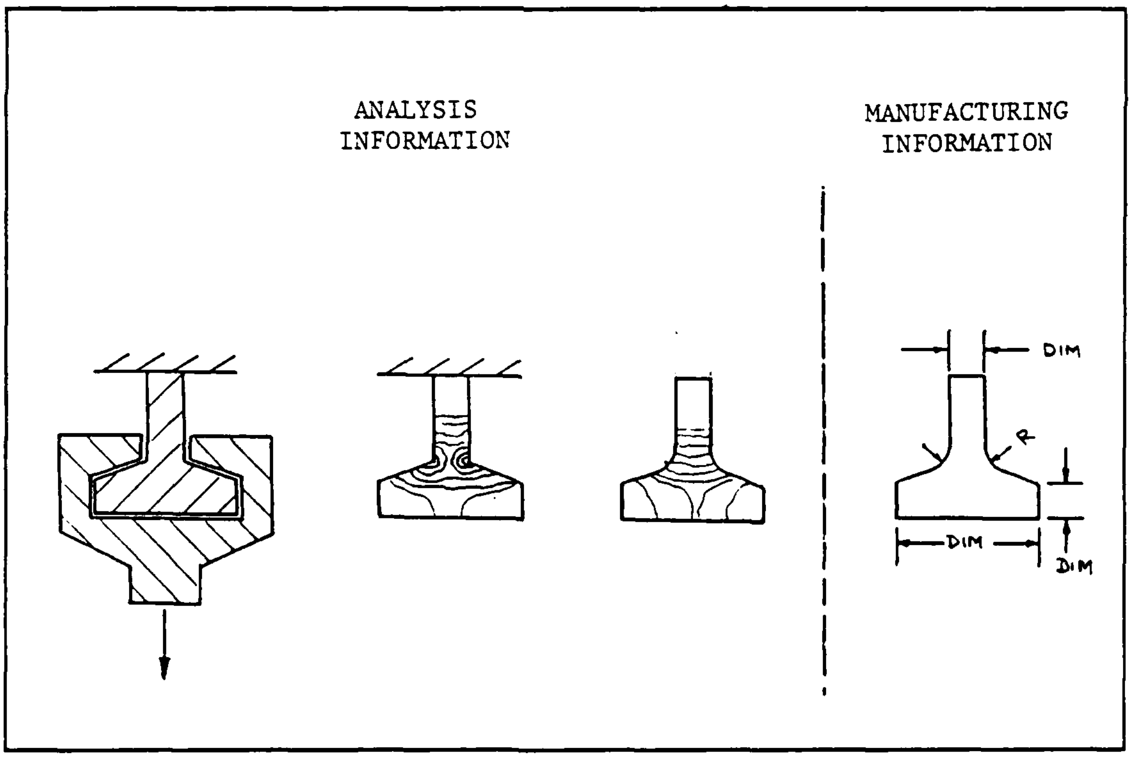


Figure E-3. An Example of Information Loss

The ULCE design process is described on two levels. The top level is described in generic system engineering and aerospace design terms. Procedure flow diagrams and data flow diagrams are used to illustrate the ULCE architecture. The one-level-down description of procedure flow and data flow is tied extensively to examples from the replacement C-130 high sink rate gear design.

2. Top Level ULCE Architecture

The top level ULCE architecture procedure flow is diagrammed in Figure E-4. Three top level procedures are defined:

- Generate Design Alternatives
- Plan Design Decision Process
- Make Design Decisions.

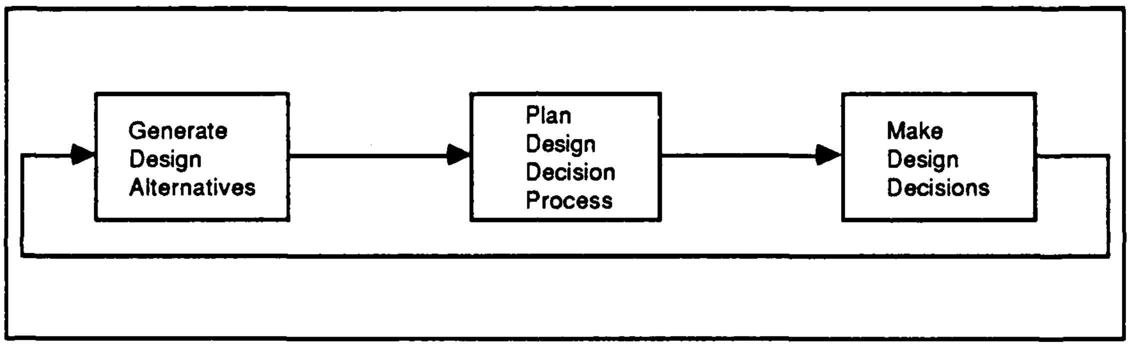


Figure E-4. ULCE Procedure Flow

a. Generate Design Alternatives

Producibility is now incorporated in designs through specific design recommendations. At Lockheed, Supportability advocates that design-to recommendations be presented to the designer in design terms. What this means is that supportability and producibility specialists must be able to develop design alternatives that are compatible with maintenance and production concepts. This requires that the proposed ULCE architecture addresses how design alternatives can be generated by engineering specialists who are currently not considered to be designers.

The proposed architecture provides an integrated process for creative generation of design alternatives by all members of the design team. This process will bring tools currently used by preliminary design engineers, landing gear designers, producibility and

supportability specialists together with systems engineering tools for interdisciplinary communication.

The Generate Design Alternatives process will allow all members of the design team to suggest design ideas. Design alternatives will include maintenance and production concepts as well as ideas for the performance-related system elements that are currently considered part of the design definition. These design alternatives should be responsive to requirements, but they need not meet all requirements simultaneously.

b. Plan Design Decision Process

Planning the design decision process involves identifying design decisions and setting up a timetable for making those decisions. The design decision process plan must consider the implications of all the relationships among attributes of the design alternatives. For relatively simple designs, this can be done by a small group of design integrators. Although the process of design decision planning is not usually considered explicitly for relatively small-scale efforts, the ability to understand the implications of a complex set of interacting requirements and design attributes is a mark of design genius.

The effectiveness of an individual design genius in integrating the efforts of a large group of engineers on a complex, multidisciplinary problem is limited by program schedule constraints. Lockheed's approach to the ULCE architecture is to provide computer support for identifying and scheduling the design decisions that effect integration of the design. This is the *Meta-Design* concept. Software support is used to decompose a symbolic representation of alternatives for attributes of the design concept into discrete design decisions and interfaces between those decisions. A decision timetable can be developed by considering the amount of time required to apply simulation and analysis tools to evaluate the alternatives. In fact, this Meta-Design, the Plan Design Decision Process, is a procedure for defining and building the next step of the design process itself. It plays a principal role in facilitating the design optimization capability of the proposed ULCE architecture.

c. Make Design Decisions

The design decision process plan will identify design decisions involving a mixture of qualitative and quantitative considerations. Design decisions are made interactively, using an integrated tool kit of knowledge-based systems, design optimization, and theory of measurement techniques. The design task computing environment must capture

substantiation developed by the design team to support design decisions. This substantiating information will be used for Preliminary and Critical Design Reviews and to support iteration with other decisions when required.

The design decision-making tasks will specify an alternative for each of several design attributes. If alternatives can be specified that fully meet requirements, the design team moves on to other design decisions. If no suitable alternative is found, the impact of prior decisions on the current decision is assessed, and these prior decisions are iterated based on this assessment.

3. ULCE Data Flow

The flow of data, or more properly information, at the top level is shown in Figure E-5.

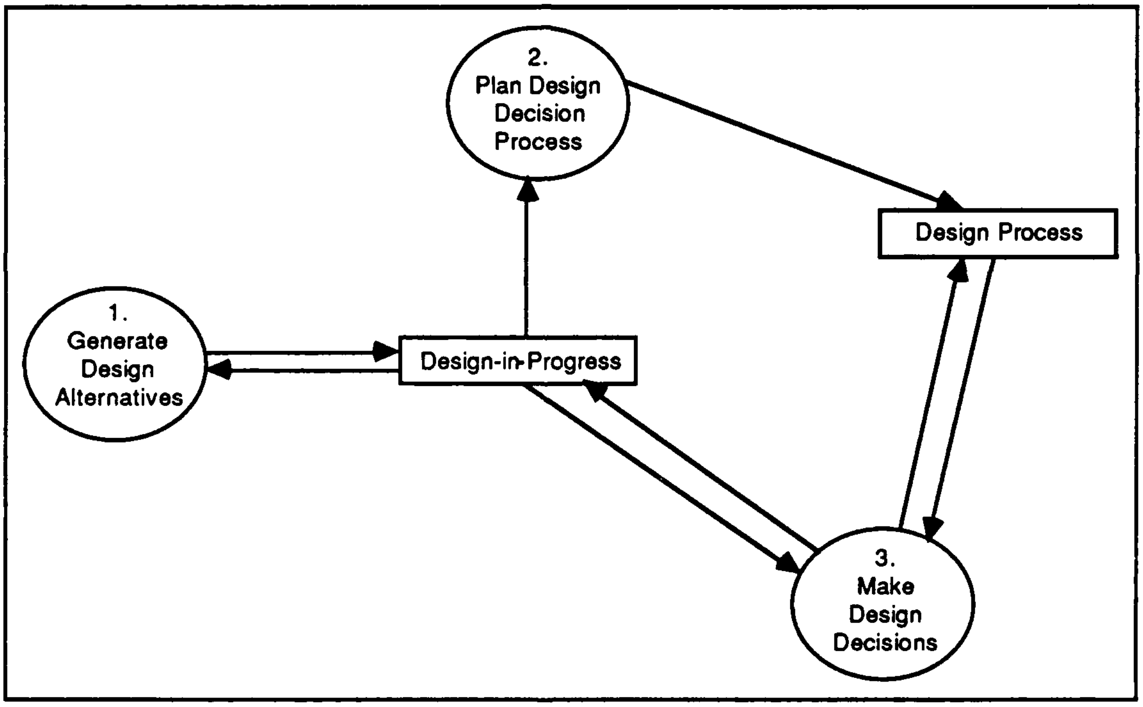


Figure E-5. ULCE Data Flow

Procedures are represented on the data flow diagram by bubbles around the name of the procedure. These names are referenced in the procedure flow diagrams. Design tools are shown on the design diagram by rectangles around the name of the object-centered tool.

Procedures appearing on the diagrams are performed by the design team. Arrows between the procedures correspond to interactions between the design team and the design tools. These interactions include: (1) defining a design object, (2) instantiating a design object, or (3) invoking a method of a design object for some instance of the object.

Design objects corresponding to requirements, functions, and implementations of alternative design concepts are defined by the Generate Design Alternatives procedure using an object-centered representation of the Design-in-Progress. The computer programs corresponding to these design objects are accessed (as data) by other computer programs that provide computing support for the design team in planning the design decision process. Requirement, function, and implementation (system description) design objects in the Design-in-Progress contain design attributes and engineering relationships. These relationships are extracted from the design object definitions and are used to decompose the task of selecting among alternatives for attributes of the design concept into smaller design decision tasks. The design process planning procedure then instantiates each of these tasks (definition of a decision-task object is part of development of the underlying ULCE software). Instances of the design tasks collectively make up a design-process computing environment. Each instance of the decision-task object provides access to local decision support tools for use by the design team as part of the Make Design Decisions procedure and supports tracking of the global decision-making process, also part of the decision-making procedure.

a. The Design-in-Progress

The Design-in-Progress element, shown in Figure E-5 and in later data flow diagrams, is one of the most critical elements of the proposed approached. It is the core of the ULCE architecture. Some amplification and clarification of its features, characteristics, and functions should provide an understanding of its role in changing the design process and the quality of the resulting product.

The Design-in-Progress is the repository of all knowledge relating to the product being designed that has been generated in the course of the design activity. As the design proceeds through its various stages, the amount of information or knowledge about that design continues to grow. This information includes not only an increasing level of detail of product-specific descriptive information but also historical data on analyses, design decisions, etc. All of this knowledge is retained in the Design-in-Progress. Equally

important, all of the current information about a particular design is readily available to all of the design disciplines of the design team.

(1) Content and Structure

The Design-in-Progress contains design concepts. The design concepts provide a context in which names for attributes of the design have meaning. These meanings are defined procedurally, that is, the design concept also includes methods which are procedures where the attributes occur as input or output variables. For example, a design concept for a shock strut cylinder might be represented in the Design-in-Progress. The shock strut cylinder could have an attribute *cylinder-diameter*, which is used by procedures such as *compute-cylinder-volume, construct-diameter.*, *construct-CATIA-model-of-cylinder*, *draw-sketch-of-cylinder-concept*, etc.

The significance of the Design-in-Progress is that the design concepts integrate the representations used by different disciplines to describe and evaluate the design (Figure E-6). These representations appear to the design team as different views of a single object. Representation in one of the views is accomplished by methods of the design concept. Changes to one of the attributes of the design concept are made and controlled in the design concept itself. The changes are automatically reflected in all of the views.

The methods for constructing a representation (for example, a sketch) are defined as part of the Generate Design Alternatives process. Once a design concept has been defined, the representation in any one of the views can be accomplished by executing the procedure. This provides the design team with a powerful parametric design capability. It should be noted that the term "parametric" is used very loosely here, since the attributes of the design concept do not neet to be numerical in nature, and will often represent qualitatively different alternatives. Alternatives for the attributes of the design concept collectively define the alternatives developed in the Generate Design Alternatives process.

Design concepts in the Design-in-Progress can be requirements, functions, or implementation concepts. Procedures for simulating or analyzing aspects of the design are implemented in the Design-in-Progress as methods for the appropriate design concept (requirement, function, or implementation). The Plan Design Decisions process accesses all the design-concept methods that are applied as part of the evaluation (as opposed to methods that represent a view of the concept for evaluation or communication purposes). These methods are the relationships among design attributes that allow requirements to

flow through the concept. The development of the design decision process plan is based on an examination of this requirements flow.

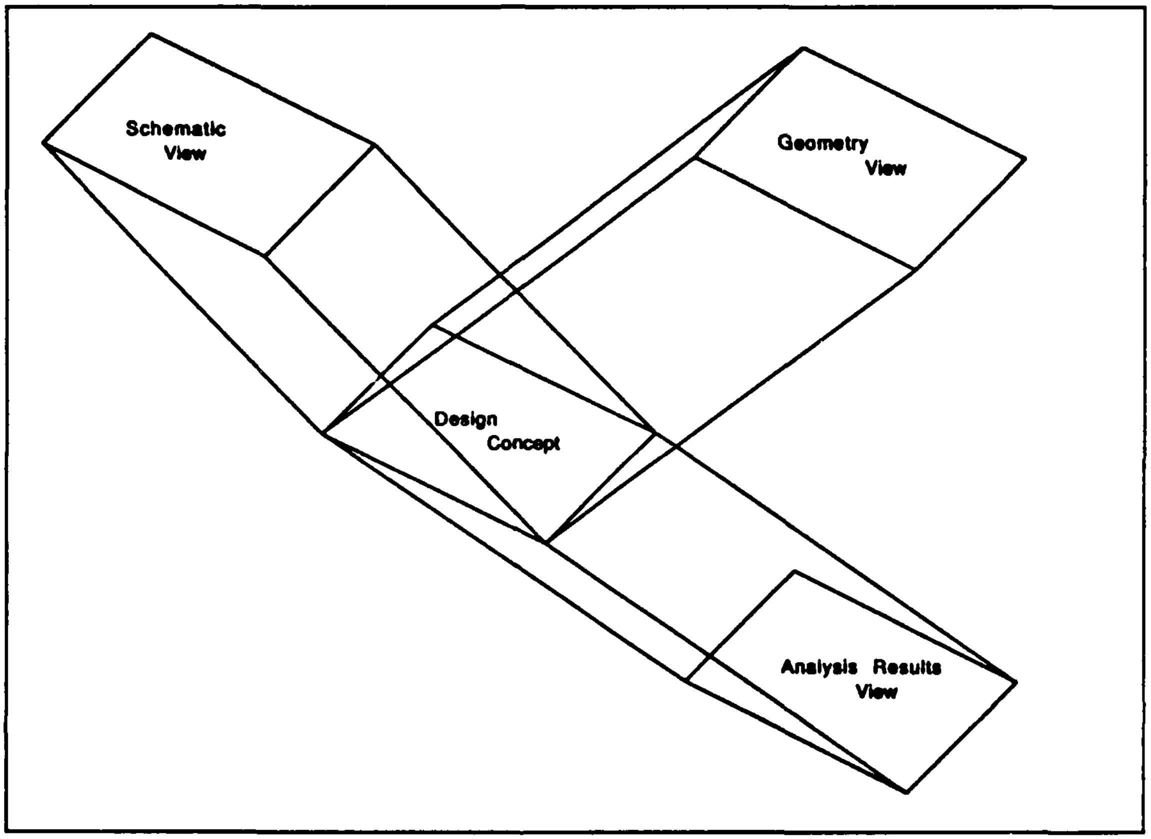


Figure E-6. Design Concepts

The design team selects particular values for the attributes of the design concepts as part of the Make Design Decisions process.

(2) Components

In summary, the components of the Design-in-Progress include the following:

- Templates for data describing a design concept. Design concepts include requirements, functions, and elements. The templates have "slots" for the design attributes.
- Specific instances of designs. The instances correspond to a template. Attribute slots are filled in with specific values.
- Two kinds of methods (procedures) are associated with the design concept templates.

1. Methods for generating a descriptive representation, such as a drawing, sketch, 3-D geometric model, finite element model, schematic, or block diagrams, from the attributes of the design (specific instance).

2. Methods for evaluating (analyzing or simulating) some aspect of a specific design instance, e.g., computer-aided analysis such as finite element or computational fluid dynamics anlaysis. Results of these evaluations become attributes of the design.

4. One-Level-Down ULCE Architecture

A walkthrough of the one-level-down ULCE architecture is described below, using examples from the replacement C-130 high-sink-rate landing gear design.

a. Generate Design Alternatives

A system engineering approach is used to generate alternative designs, including requirements flowdown, functional analysis, and system definition procedural steps (Figure E-7). Each of these procedures is executed once during each cycle through the top-level ULCE architecture.

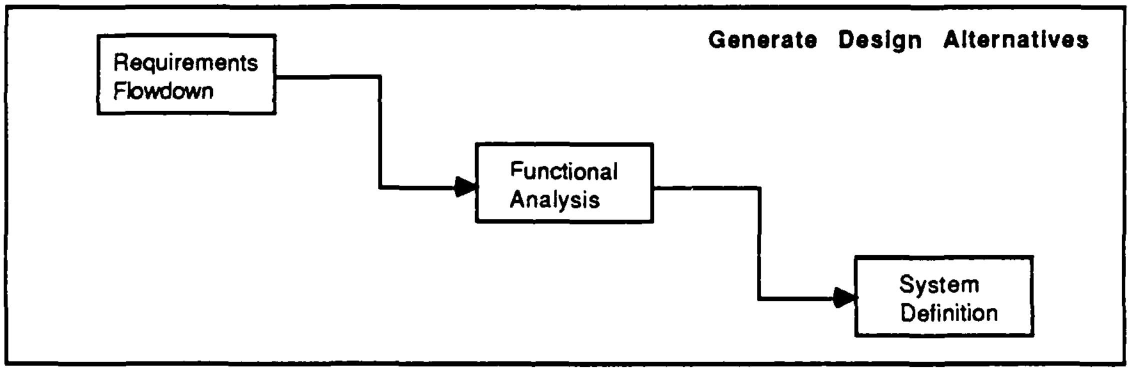


Figure E-7. Generate Design Alternatives Procedure Flow

Requirements Flowdown involves each engineering discipline (including supportability and producibility) nominating requirements derived from customer needs. For example, AirLand Battle 2000 and clandestine operational needs for the C-130 lead to requirements to land and operate on short landing strips with a minimum of surface preparation and bomb-damaged runways. These conditions necessitate reduced landing distances, implying higher glide slopes (15 feet per second sink rate at 130,000 lb. design landing weight--current C-130 landing gear was designed for 9 feet per second at 130,000 lb. and 5 feet per second at max. landing weight of 155,000 lb.), minimum flare, and improved strut damping for rough field operations. These operational needs, and others

derived from R&M 2000 considerations, will require the C-130 to be operated from small, austere locations and dispersed operation locations (DOLs) with minimal manpower, spares, support equipment, and facilities. Ruggedness for unprepared field operations, minimal preventive maintenance, rapid repair, and maintenance in a nuclear, biological, and chemical (NBC) environment also will be required.

Examination of the production facilities available and consideration of cost and schedule impact leads to a requirement to use existing C-130 forgings and tooling whenever possible.

Once these requirements are derived by the design team, they become part of the Design-in-Progress description (Figure E-8). They are prioritized into (hard) requirements, goals, and criteria.

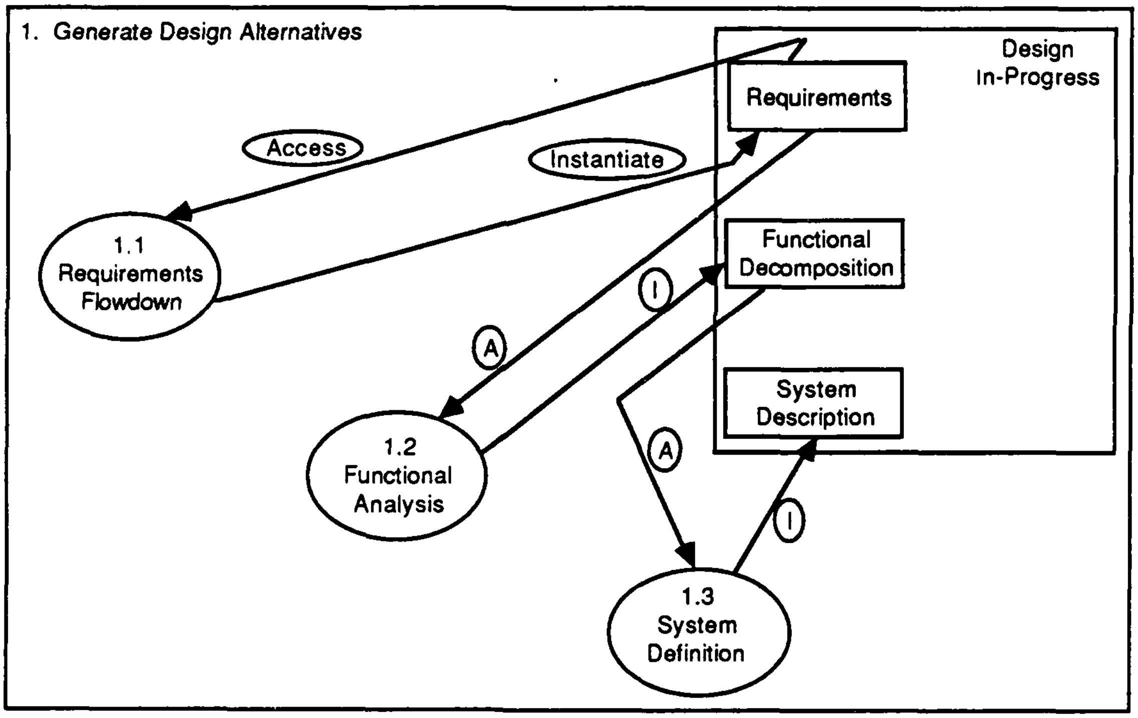


Figure E-8. Generate Design Alternatives Data Flow

Functional Analysis (Figure E-9) requires the design team to explicitly describe the functions that must be accomplished to meet the requirements. For example, functions such as *dissipate energy, control rate of strut extension during rebound, eliminate damping decay, retract gear, forge NLG cylinder, jack MLG, operate in sand, mud, grit environment* are defined at various levels of the functional decomposition (Figure E-7). Each function is associated with meeting one or more requirements, or, functions may

identify unintended couplings (with synergistic or adverse effects) among system components or describe failure modes of an implementation alternative at lower levels of the functional decomposition..

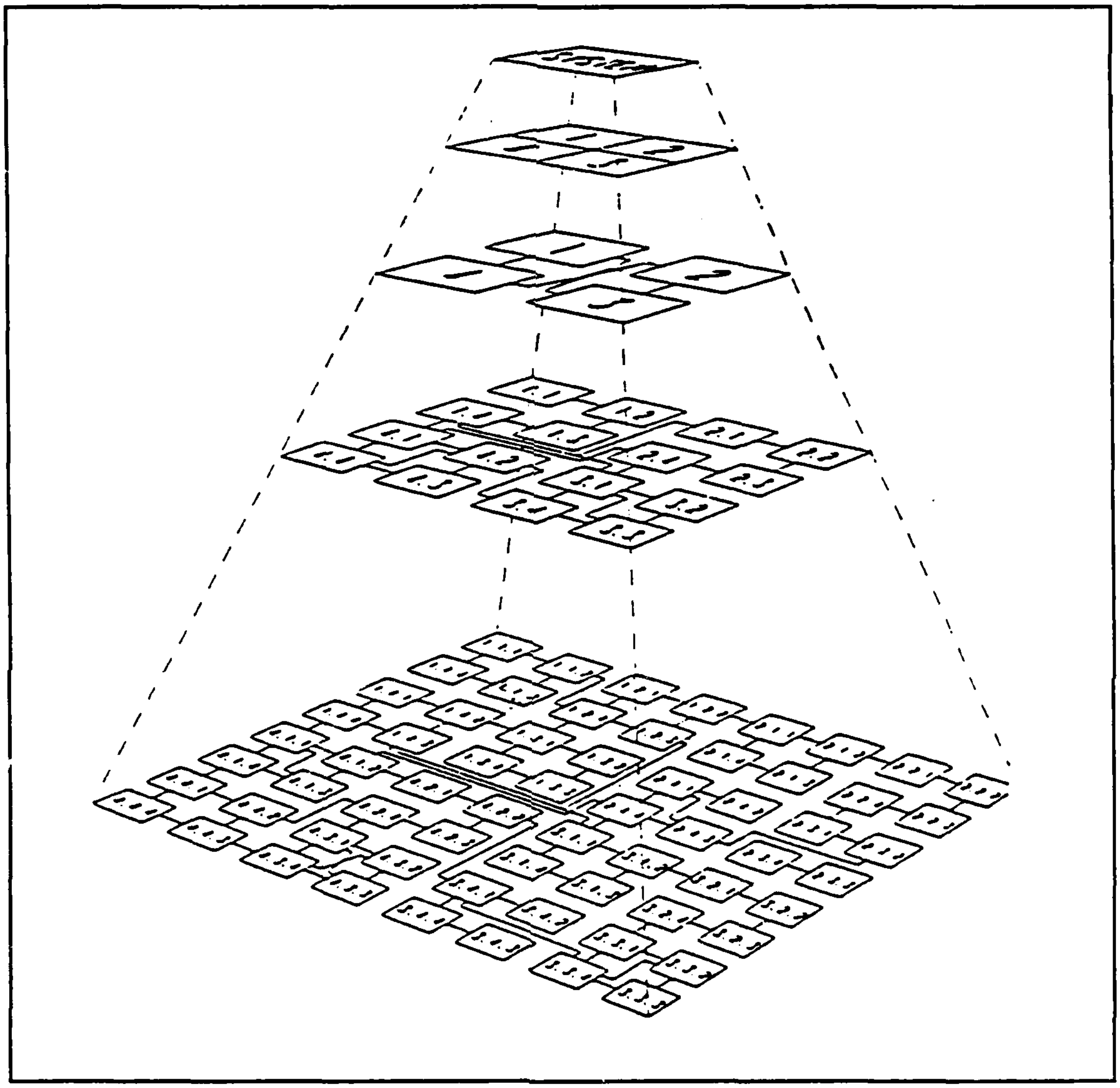
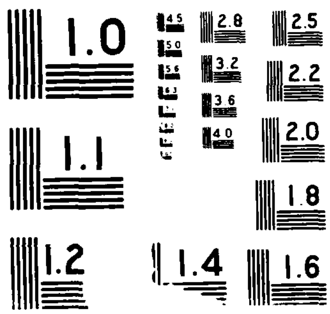


Figure E-9. Functional Analysis

Several alternative functionalities are defined to meet a requirement (whenever possible). A requirement to reduce strut binding in the C-130 MLG may be addressed by some combination of alternative functions such as *constrain piston deflection, maintain wheel in vertical position under load*, or *lubricate strut*. Objects corresponding to each of the alternative functions are created and included in the Design-in-Progress description (Figure E-8). Alternative implementations for each of the functional alternatives then are

AD-A194 516 ARCHITECTURE AND INTEGRATION REQUIREMENTS FOR AN ULCE (UNIFIED LIFE CYCLE (U) INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA M L BREI ET AL. APR 88 IDA-P-2063 2/2

UNCLASSIFIED IDA/HQ-87-32844 MDA903-84-C-0031 F/G 5/1 NL


1.0
1.1
1.2
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

invented by the design team (System Definition--Figure E-7). In the high sink rate C-130 landing gear, a long-stroke shock strut has been implemented to dissipate the energy on landing (Figure E-10). A floating separator piston separates air and oil in the shock strut,

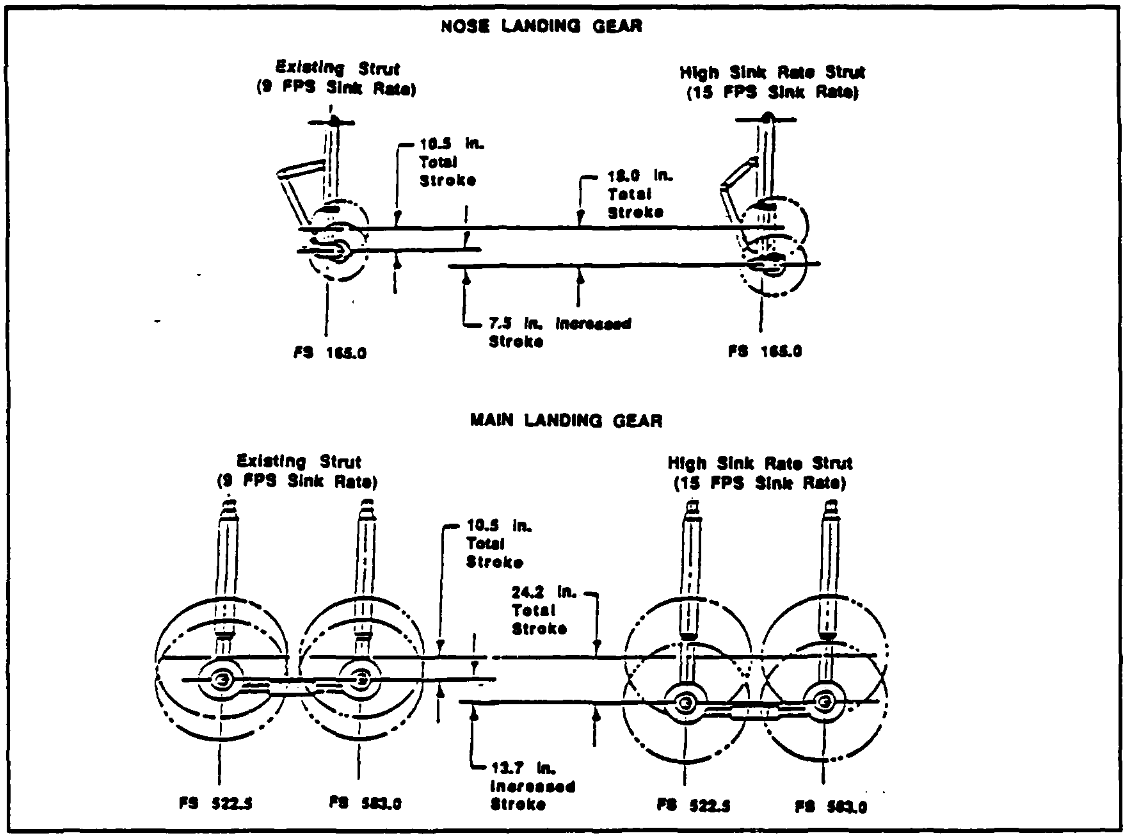


Figure E-10. Comparison of Strokes of Existing and High Sink Rate Struts

preventing emulsification of the oil (Figure E-11). Lubrizol No. 1395 (zinc dithrophosphate) and Emerest 2301 (metholo");eate) have been added to the MIL-H-5606 strut fluid to increase lubricity for reduced shock strut binding. A manual decoupler placed on the horizontal torque shaft of the main landing gear and a second hydraulic motor and brake assembly at the aft-angle gear box location allow the longer stroke high sink rate gear to be independently retracted for jacking (Figure E-12). These implementation alternatives are identified in the Design-in-Progress by an alternative system hierarchy breakdown similar to that shown in Figure E-13. Other representations also will be available, such as the three-dimensional parametric geometry descriptions which are programmed as methods for the appropriate objects.

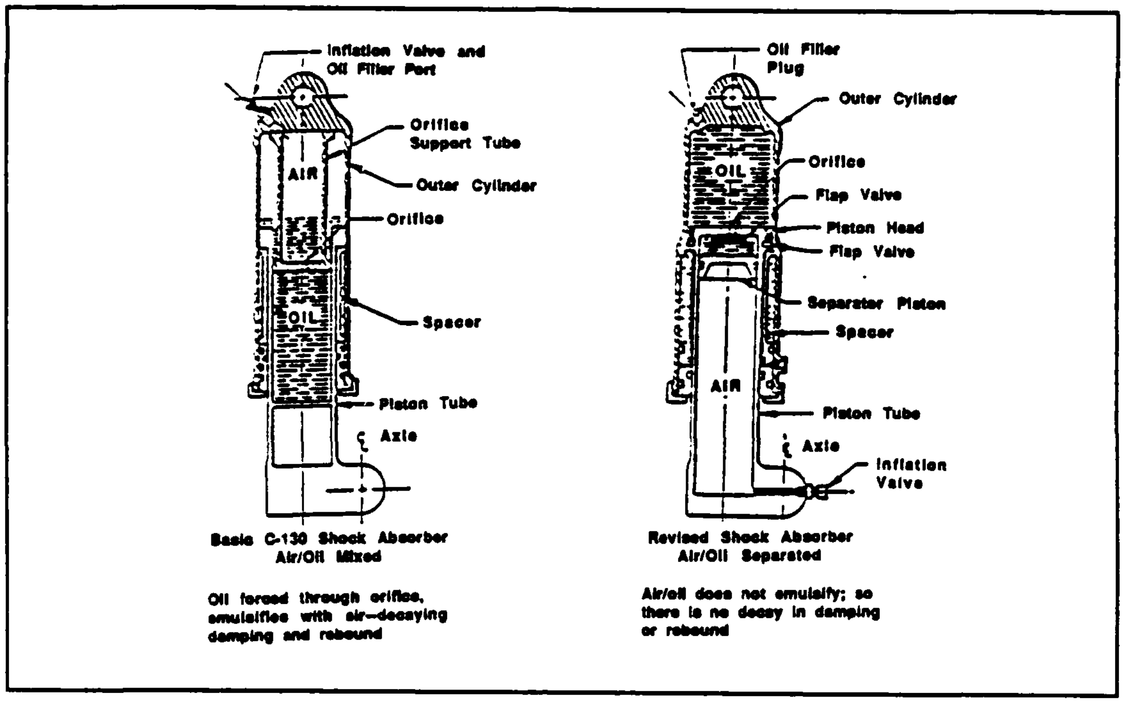


Figure E-11. Gear Bottoming Prevented

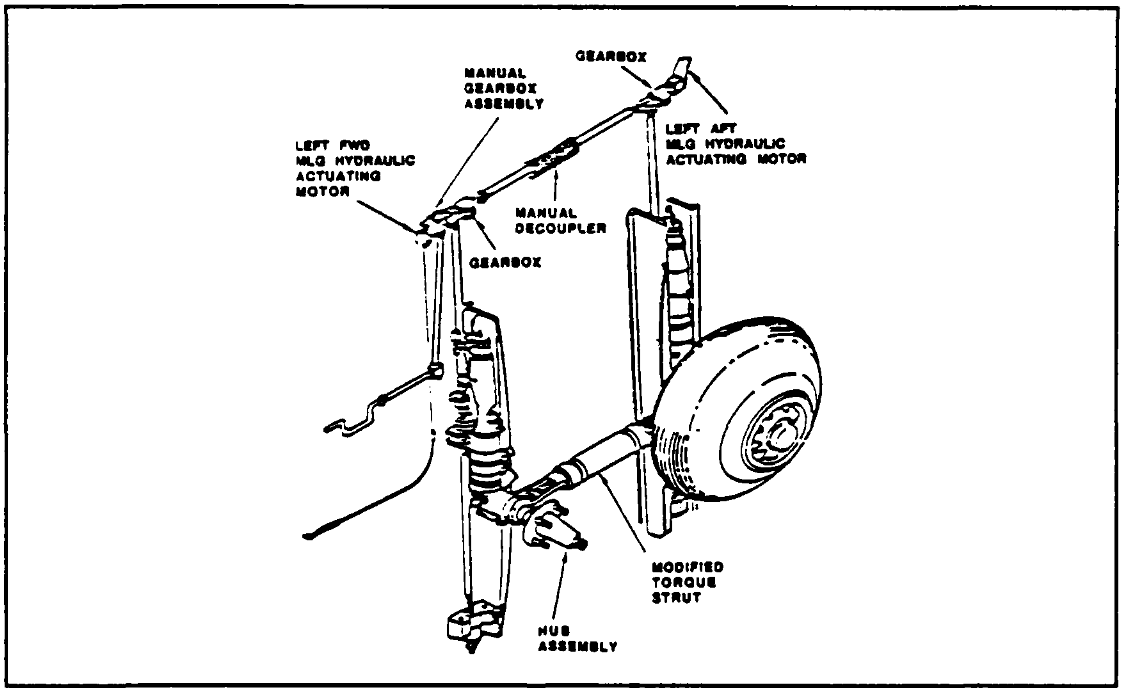


Figure E-12. MLG Minor Modified and Hub Assembly

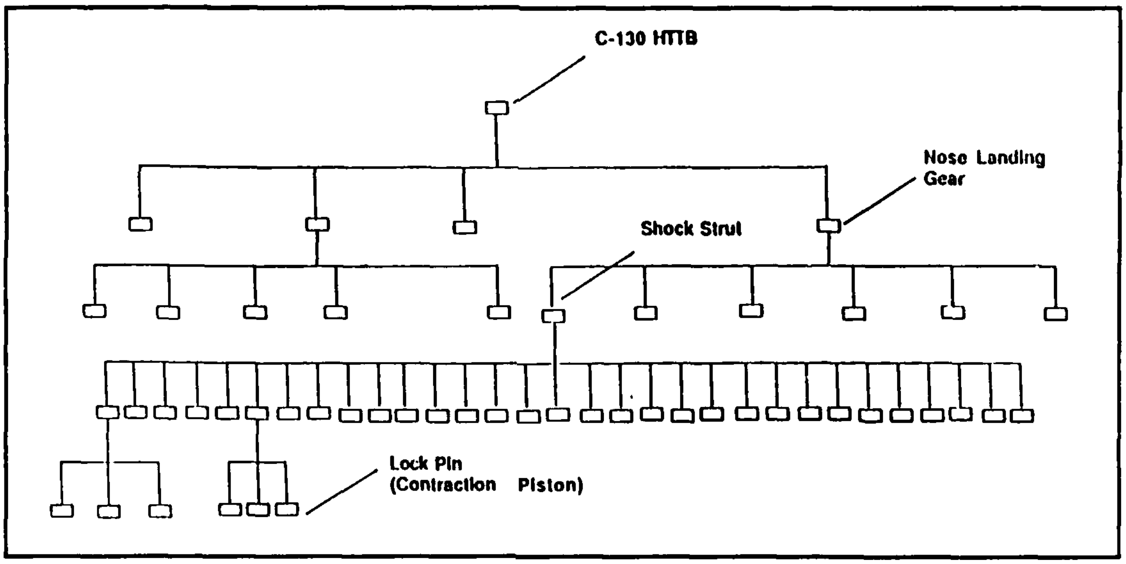


Figure E-13. System Definition

The translation of supportability and producibility requirements into design terms is made in the System Definition stage. This approach--translation into specific design concepts--has been put forward by supportability engineers at Lockheed as a solution to the problem of balancing performance, schedule, and cost against supportability considerations. Indeed, the approach is similar in some ways to the current practice of *incorporating producibility considerations into the design.*

The proposed ULCE architecture takes this approach one step further and includes an explicit step (design decision process planning) to identify conflicts and opportunities implied in an integrated Design-in-Progress description. This step contains supportable or producible alternative design concepts as well as design concepts driven by performance, cost, and schedule considerations.

b. Plan Design Decision Process

A walkthrough of design decision process planning for the landing gear design is based on examples from the Multilevel Optimization by Linear Decomposition (MOLD)--a computer program developed at Lockheed. Although the basic procedures outlined in Figure E-14 generally are applicable in any ULCE architecture, the techniques for providing computer support for these procedures as described in this example are specific to MOLD and should not be viewed as constraints on the ULCE architecture.

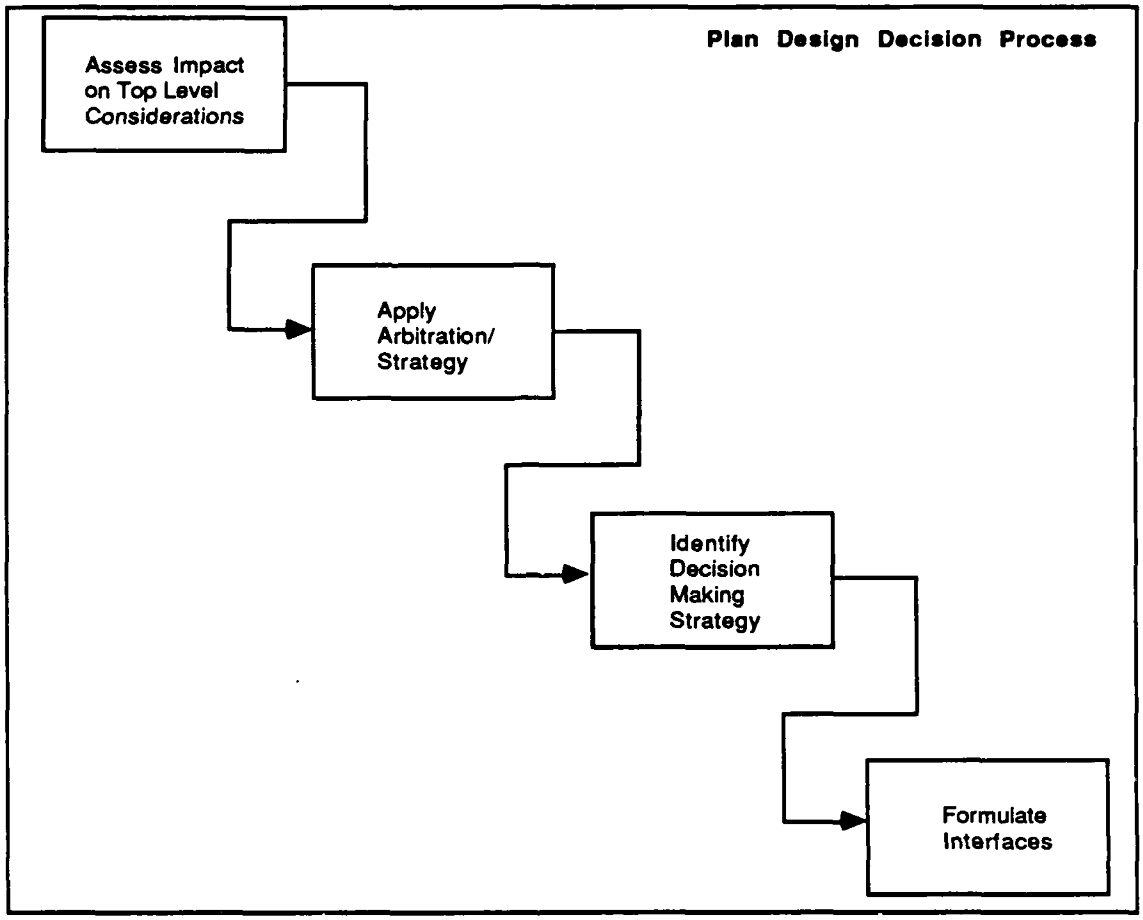


Figure E-14. Plan Design Decision Process Procedure Flow

Top-level design considerations, defined in the Design-in-Progress environment, are accessed to use in planning the design decision process (Figure E-15), and the effect of each attribute is assessed. In MOLD, this is accomplished by constructing a network representation of the attributes and their interrelationships. Top-level design considerations appear on the network as required or goal attributes for the design. The impact is assessed by examining connectivity in the attribute/relationship network. The detail of a portion of such a network that has been developed for landing gear design is shown in Figure E-16. In this example, aircraft parameters that directly effect the landing gear design (landing-speed, k-e-aircraft, aircraft-type, v-sink, and so on) are placed at the top level. *Landing-speed* is used to compute *stop-time*, as indicated by an arrow connecting these two attributes. Thus, *stoptime* is an attribute that directly impacts the top-level design considerations.

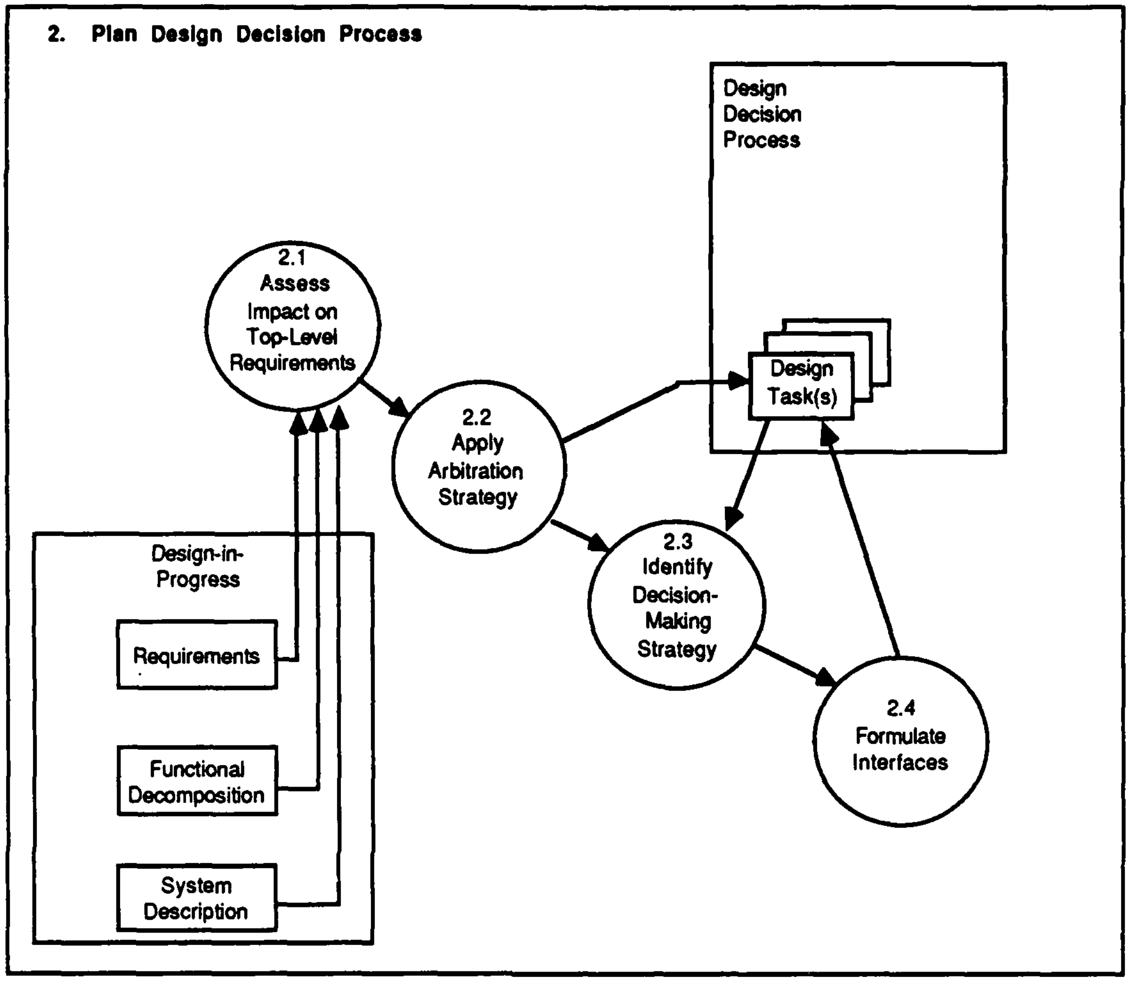


Figure E-15. Plan Design Decision Process Data Flow

Once the effect is assessed, heuristics corresponding to scheduling and arbitration strategies are applied to group the attributes of the design and to identify a decision timetable. Again using MOLD as an example, attributes that directly impact the top-level considerations are decided first, and those that have a less direct effect are specified in subsequent design decisions. MOLD schedules the design decisions based on the strategy that all attributes that have an equal impact on the top-level considerations should be determined at a single stage of the decision process. Attributes that are connected within a stage are specified as part of a single decision-making task. The process of grouping attributes into design decision-making tasks using MOLD results in a smaller *decision-task network*. The decision-task network for the design process, (Figure E-16 is a part of this network), is shown in Figure E-17. The numbers at each decision point represent the scheduling assigned by MOLD.

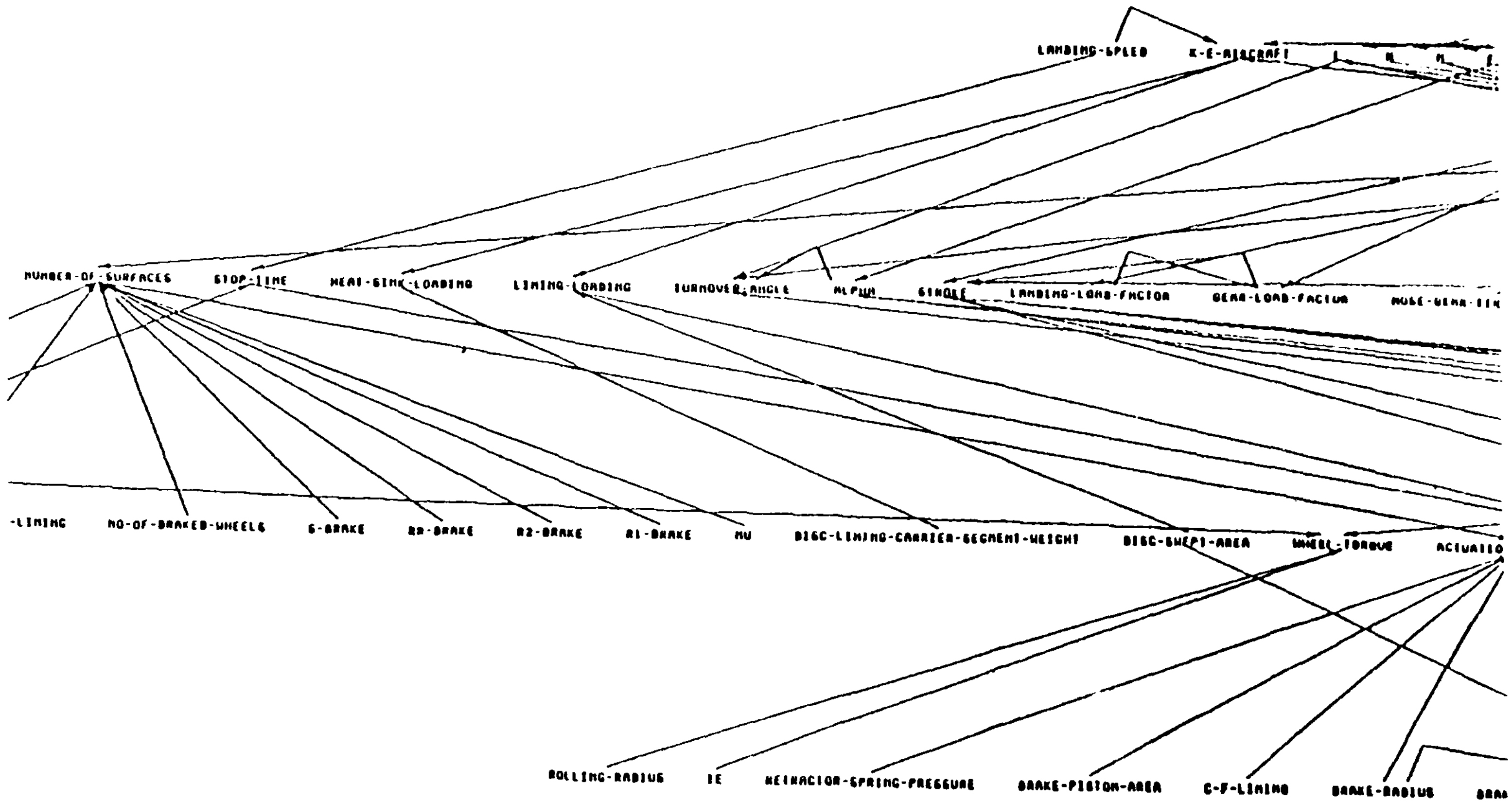


Figure E-16. Landing Gear Design

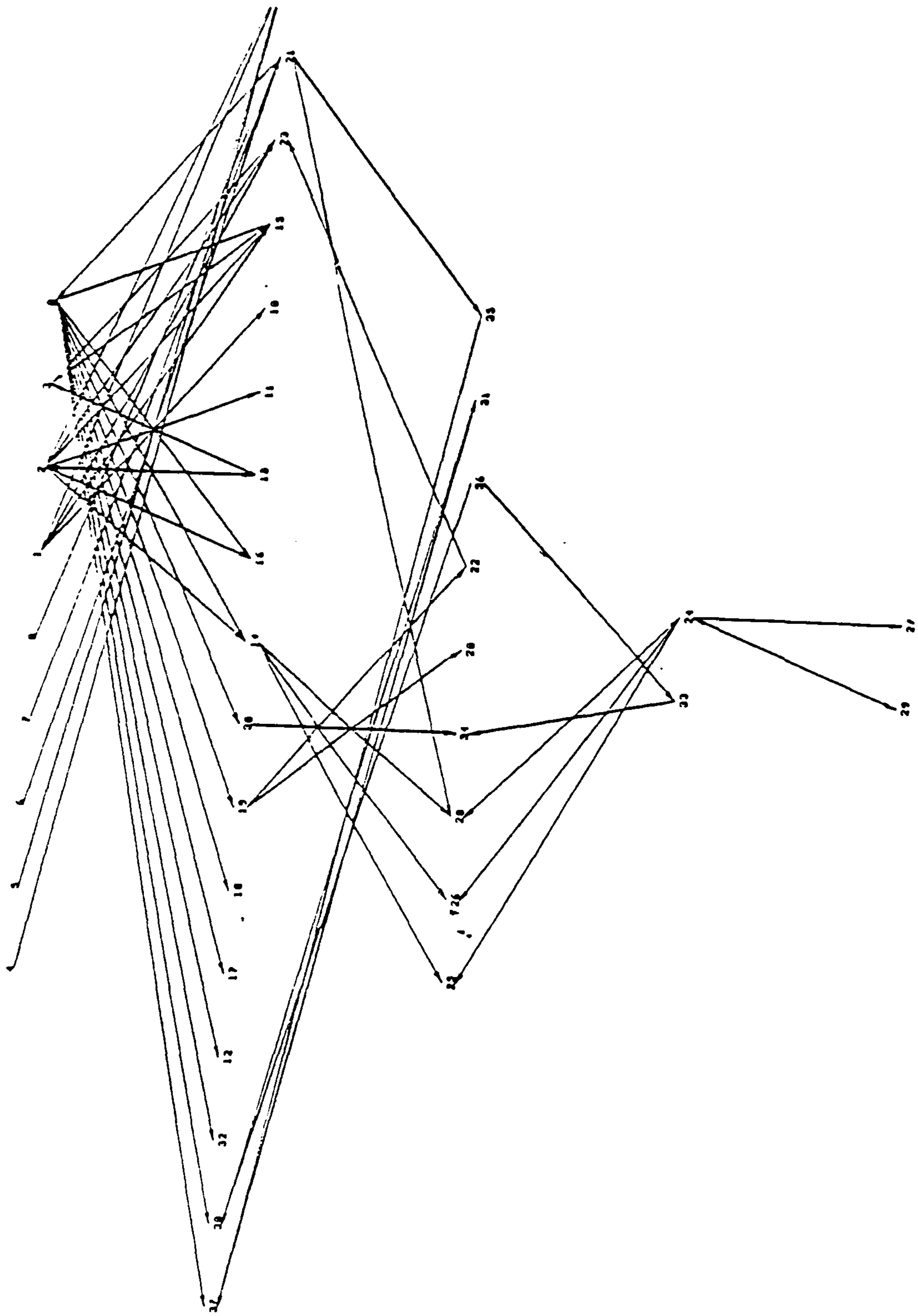


Figure E-17. Decision Task Network

Once the tasks have been identified, based on scheduling and arbitration decomposition heuristics, design-task objects are instantiated in the Design Decision Process (Figure E-15). A decision strategy is identified for each task. To allow iteration, interfaces between tasks are defined, and these will be formulated as part of the Make Design Decisions procedure, as conflicts between decision tasks emerge.

Based on the decision strategy and on estimates of the number of engineering hours required to evaluate each design alternative, a detailed decision schedule is developed based on the decision-task network.

c. Make Design Decisions

Procedures for making design decisions in the ULCE architecture must be defined to allow integration of qualitative and quantitative decisionmaking. An approach to accomplishing this is illustrated in Figure E-18. Development of this approach began by considering the steps performed in an interactive quantitative optimization process. These steps are compared with a qualitative decision process to identify common procedures. In the resulting ULCE procedure flow, alternatives are identified by combining ideas that were developed in the Generate Design Alternatives process to come up with integrated designs that are responsive to a broad range of requirements. For example, features of the C-130 major supportability mod (Figure E-12) might be combined with the long-stroke strut and other features of the basic high sink rate landing gear design concept. Identifying these alternatives can be accomplished very efficiently using design tools in the Design-in-Progress environment.

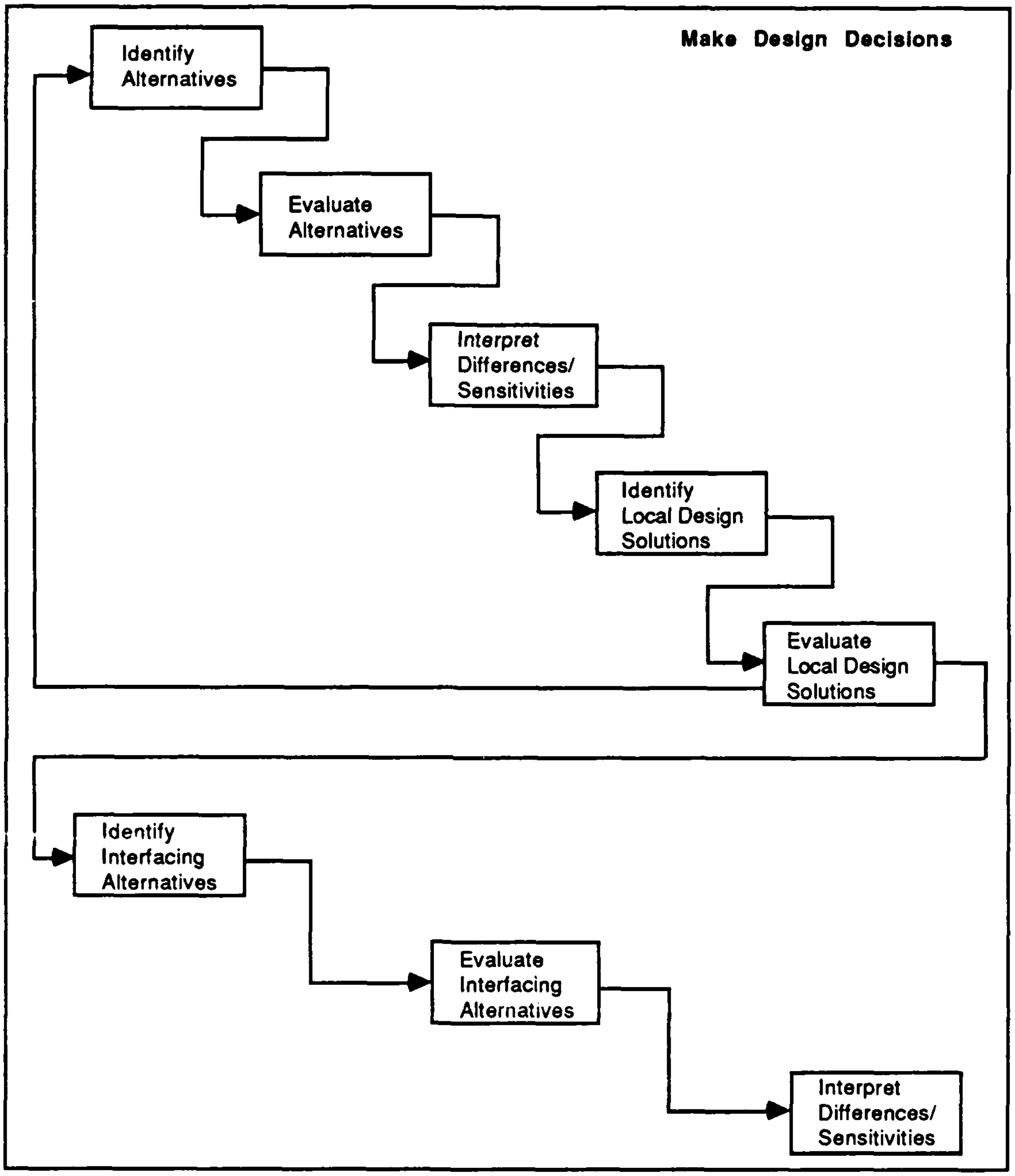


Figure E-18. Make Design Decisions Procedure Flow

The design team evaluates the alternatives using the evaluation methods that were built into the design concept when it was first represented in the Design-in-Progress environment. These methods are invoked by the design team through the appropriate piece of the functional or implementation (system description) breakdown. Thus, the design concepts function as a kind of user interface and executive for the evaluation methods. The *dissipate energy* function is linked to engineering data and test/analysis methods to evaluate shock absorber efficiency, and so on.

The details of the interaction between the design team and the Design-in-Progress are shown in Figure E-19. Prototype alternative design concepts residing in the Design-in-Progress are accessed in the process of identifying alternatives. The process of evaluating alternatives involves making specific instances of the design concept (designs). One of the differences between the design concept template and an instance of the design concept is that the template has only slots for the attributes of the design concept while specific values for these attributes are associated with an instance of the design concept. The design team selects particular values for these attributes at this stage of the design process. The specific instances or designs are the representation in the Design-in-Progress of the alternatives to be evaluated. Methods to analyze (or simulate) an aspect of the design are invoked and executed as part of the evlauation process. Since these methods are the same for each instance, they are associated with the design concept itself. Each instance has the ability to access these methods, to apply them using its own values for design attributes, and to set values for other design attributes using the results. These analysis and simulation tasks are performed by members of the design team as part of the Evaluate Alternatives process. Results of these analyses are built into the representation of the design alternatives in the Design-in-Progress. These results are accessed by the design team during the Interpret Differences/Sensitivities process.

The design team next applies creative engineering judgment to interpret the differences between the design alternatives as indicated by the evaluation results or to quantify the sensitivity of the design evaluation results to changes in the alternatives. Some combination of qualitative interpretation and sensitivity analysis will be required for most design decisions. For example, the effect of using strut fluid additives on weapon system repair-kit cost and on weight would be quantified, or the effect of partial hydraulic retraction of the landing gear on possible failure modes and criticality would be assessed in qualitative terms. Results of these sensitivity analyses and interpretations are captured in the Design

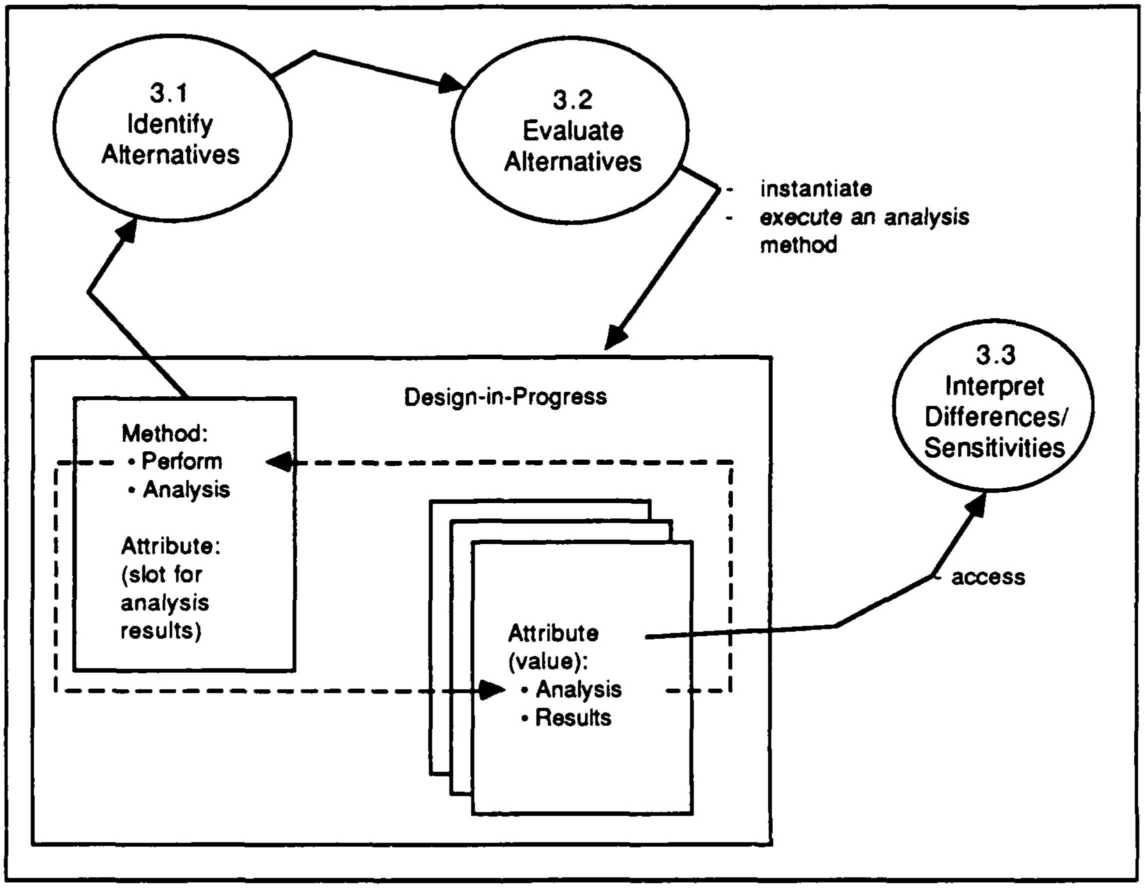


Figure E-19. Alternatives

Decision Process object-centered environment (Figure E-20) to document Contract Data Requirements List (CDRL) items and for use in iterating the decision process.

Based on the interpretation/sensitivity analysis results, local design solutions are identified. These local solutions represent a combination of the available alternatives that should be satisfactory, based on evaluation of similar alternatives. The local design solution may differ in significant respects from the design alternatives that have already been evaluated. The proposed local design solution is then evaluated, and if it is not satisfactory, the process is iterated (Figures E-18, E-20).

All but the first design decisions are based in part on the results of previous decisions in the design decision task network (Figure E-17). If a satisfactory solution cannot be found to the current decision task, alternative outcomes to previous decisions (interfacing alternatives) are identified and evaluated. Based on these evaluations, the effect that changing the outcome of previous decision tasks would have on the current task is

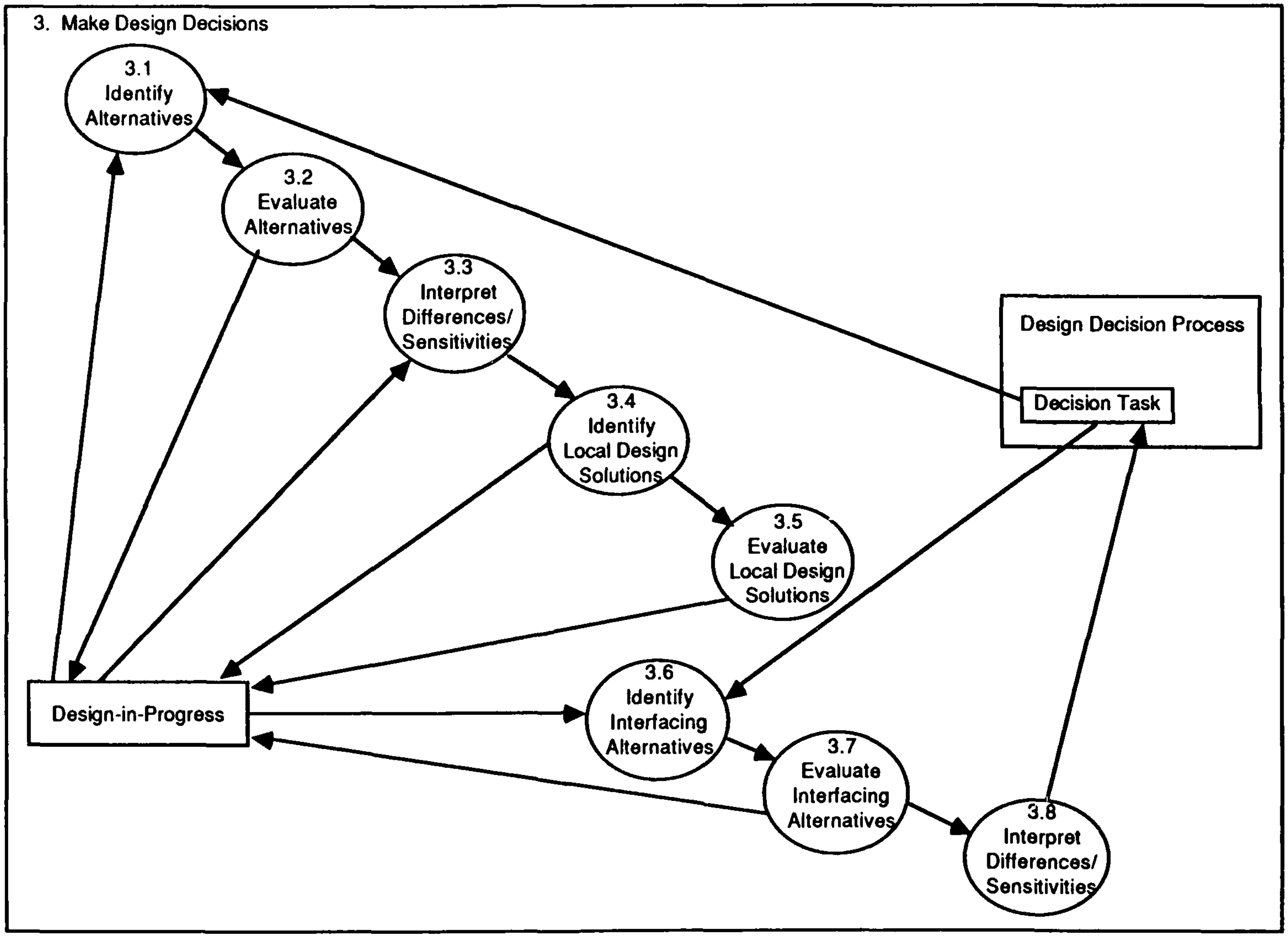


Figure E-20. Make Design Decisions Data Flow

interpreted (what if ? analysis), and those decisions are iterated subject to the constraint that their outcome now must improve the feasibility of downstream decisions.

5. Technology Approach to the Proposed ULCE Architecture

If the proposed concept of the ULCE design process can be fully implemented both in terms of integration and automation, a quantum improvement in the efficiency and productivity of a design team and in the quality of resulting products can be expected. The current crop of proven hardware and software elements are not capable of adequately supporting the computing needs of the proposed ULCE architecture. However, there are a number of new technology tools that are rapidly evolving and show promise in meeting the ULCE requirements. Two of these new technology tools directly affecting the proposed ULCE architecture are discussed in the following section. An expanded view of the new technology tools is included in Section F on Software Requirements.

a. Symbolic Computation and Object-Oriented Programming

Symbolic computation allows the design team to completely describe the design concept at all levels of detail in a computer-based representation. Complete description of the design involves a complex framework of names; contexts; relationships among and between named objects; and complex, highly intuitive constructs that include both data and procedures. The framework for the design description becomes more detailed and complex and is frequently revised during the design process as additional knowledge about the design problem at hand is acquired.

Computing technologies currently applied to design follow the tradition established by John von Neumann and are based on a structured yet linear algorithmic approach to software design (such as FORTRAN subroutines). They require the developer and user to translate descriptions of design concepts into an artificial representation. In contrast, symbolic computation allows the engineers developing the design description to capture a complex framework of design ideas in its entirety and to apply computing power to search, control, track, translate, calculate, draw, and simulate the design as it progresses.

Symbolic computation, as first implemented in the computer programming language Lisp, also provides a facility to write computer programs that manipulate data and/or other computer programs. Recognition of the problems associated with applying this capability to develop useful computing tools has resulted in an *object-oriented* approach to software

design. The object-oriented approach to computer programming has tremendous potential for development of computer-based engineering design descriptions.

Previous studies have recommended an object-oriented approach for the effective development of future engineering environments. A feasibility study conducted for the Engineering Information System (EIS) program, a major U.S. Air Force VHSIC effort, concluded that an object-oriented approach is necessary to implement integrated environments which meet the needs of the increasingly complex engineering process. Prototypes of the EIS will be object-oriented.

Object-oriented software design provides developers with a strategy to control computer programs that write other computer programs. The developer writes template programs called objects. The objects correspond to concepts that are used to solve the problem at hand. For example, in designing an aircraft, it is useful to have the concept of a landing gear in mind. Thus, using object-oriented programming, a landing-gear object would be defined.

Objects are given features of the concept that we are trying to capture. One approach to object-centered design provides the developer with two basic kinds of features that an object may have: parametric attributes and methods. Attributes of a landing-gear object would be *landing-gear-load-factor, shock-absorber-type (gas/oil, rubber, liquid-spring leaf-spring, coil-spring, ring-spring)*, and so on. Methods for a landing-gear objec' would include *compute-landing-gear-load-factor, select-shock-absorber-type, draw-shock-absorber*, etc. Attributes correspond to value-based characteristics. Methods correspond to functions.

The objects are typed templates that are used to create "instances" of the objects. The instances have specific values for the attributes and can execute the methods. The landing-gear object corresponds to the concept of a landing gear that an aircraft designer might have in mind. An instance of the landing-gear object is a computer-based description of a specific landing gear design.

Object-centered software design gives the design team a tool for recording and communicating engineering ideas. Extensions to the basic object-centered concept, such as the inheritance of methods and the ability to mix simple objects together to define complex objects, further enhance the usefulness of the object-centered approach for engineering design.

All of the capabilities outlined above for engineering design applications of symbolic computation and object-centered programming have been demonstrated using existing technology. Initial evaluations of object-centered programming environments for engineering design have indicated a reduction in product development costs by a factor between 4 and 12. The economic impact of these savings in design cost will undoubtedly dictate extensive application of object-centered programming technology by engineering firms surviving in the competitive business environment of the mid-1990s. For this reason, object-centered tools for engineering design can play a key role in the architecture and integration of a Unified Life Cycle Engineering system.

b. Integrated Qualitative/Quantitative Optimization

Optimization is a process of applying goals and criteria to identify preferred design alternatives meeting requirements. The optimization process presumes that quantitative or qualitative evaluations can be made regarding the suitability of the alternatives. An optimization algorithm is a method for conducting a systematic search through the alternatives to identify one or more preferred candidates.

The end product of a successful optimization process is not a design. Successful optimization results in the design team understanding the technical issues underlying the design problem, specifically trade-offs and risks. Any optimized designs that are developed are by-products. The final design decisions are always made by the design team based on the understanding they have gained from the optimization studies.

In order to be successful, the optimization process must generate explainable results. This does not imply that all steps have to be performed manually, or even that each step must be transparent, i.e., the explanation of the process need not follow the same lines as the optimization process itself. The optimization process should accumulate and organize information that contributes to the design team's understanding of the technical issues.

In order to support the design team's efforts to understand the technical issues, optimization should be interactive. The optimizer should first propose a solution to the design problem. Many algorithms for quantitative optimization use a strategy of concurrent search and approximation to explore a design space delimited by implicit constraints. A simple form of interactive explanation presents these approximations to the user through a graphical interface. The user can trace these functional relationships back to the design specification to gain an understanding of how critical aspects of the design concept interact

with each other. An extension of this approach would allow the designer/user to substitute alternative approximations for those accumulated by the optimization algorithm--a kind of "what if ?" analysis.

If all the relevant considerations can be quantified, the interactive optimization environment provides the design team with tools for examining complex alternatives and arbitrating among conflicting requirements. The integration of heuristic methods to handle discrete parameters and constraints is an evolutionary development.

It seems clear that the ULCE decision support environment will demand techniques for addressing design considerations that are not readily quantified. A combination of interactive decision support and knowledge-based systems technology appears to offer the best chance of addressing qualitative optimization. The integration of these techniques with quantitative design optimization will provide the basis for making design decisions in the ULCE architecture.

6. Application of the ULCE Architecture

The design process is naturally sequential and, of necessity, iterative. The advent of ULCE will not change that. What ULCE will change is the recurring loss and regeneration of design information that presently occurs as a product evolves through the iterative and sequential stages of the process. And, the key to that change, among others, is the complete product description and knowledge base contained in the Design-in-Progress.

There are two other major features of the proposed ULCE architecture that also can be attributred in large measure to the Design-in-Progress concept--a significant reduction in the design cycle time and a reduction in the number of design errors in the finished product.

The sequence of the top-level ULCE design process is shown in Figure E-21 using the conventional phases of concept development, preliminary design, and detailed design.

Fundamental to the ULCE concept is the understanding that, regardless of the design stage, all facets of the product design appropriate to that stage are readily available to all participating design disciplines, i.e., performance, supportability, producibility, cost, etc. Indeed the generation of design alternatives can and will include variants and/or producibility considerations. Likewise supportability and producibility factors will be input to and may significantly modify the output of the Plan Design Decision Process. And, the

design goals, criteria, and hard requirements for all of the disciplines, with the desired weighting factors, will be used in making the design decisions.

In Figure E-21, the design is shown pictorially as proceeding through the process in discrete phases with the product definition, the Design-in-Progress, being passed intact from one phase to the next. That representation is artificial in that there is no need to pass an entity from stage to stage. The sequence might better be described as proceeding down through the design hierarchy with increasing design detail being evaluated and added to the Design-in-Progress at each succeeding level. A key feature of the proposed ULCE architecture is that the overall process--Generate Alternatives, Plan the Decision Process, and Make Decisions--is repeated continually, unchanged as long as design activities continue.

An ancillary but important feature of the proposed architecture is that it has the flexibility to evaluate changes to the requirements or the design at any stage. A change in requirements that might impact the conceptual design can be evaluated even when detailed design is underway. The process will allow the designer(s) to quickly identify the functional and physical design features down through the hierarchy, from system level to component level if desired, that will be affected by the change.

The Design-in-Progress will exist throughout the life cycle of a product. It will retain all of the information and knowledge about the product including design decisions at the very first stages of concept development. Redesign for later modifications or product improvements, even 20 years after development, should pose no more difficulty than was encountered in the original design activity.

Because ULCE tools will undoubtedly develop and evolve slowly, acceptance by designers, particularly in a team environment structured for concurrent design activities, should not pose the major problem in the application of ULCE. The greatest need is the technology required to support the development of software tools far more complex than any CAD/CAE tools that exist today.

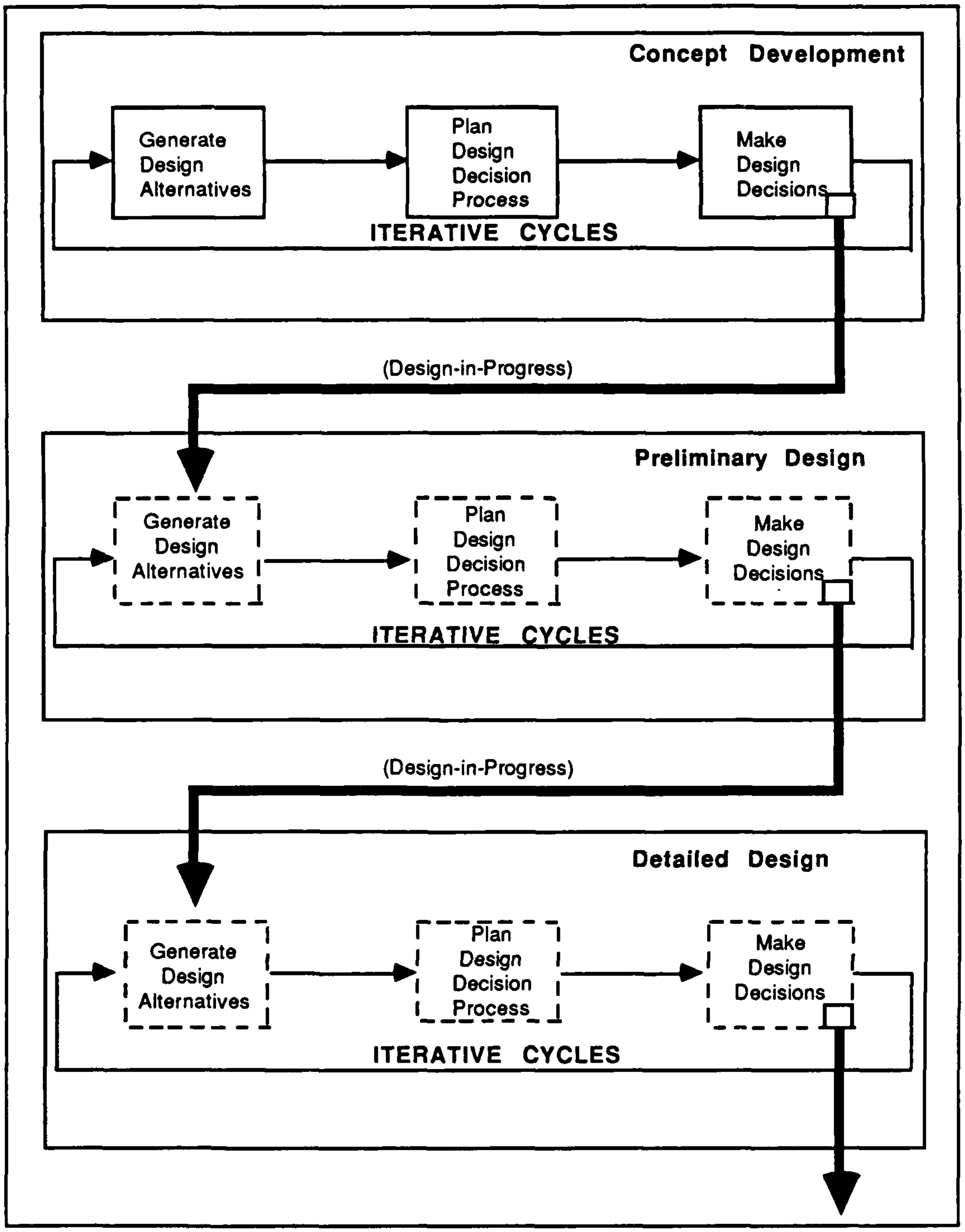


Figure E-21. ULCE Design Process Sequence

F. SOFTWARE REQUIREMENTS FOR THE ULCE ARCHITECTURE

1. Introduction

This section describes the software needed to implement the ULCE architecture described in Section E. Current design engineering environments and automated tools do not fully meet the needs of the ULCE concepts. There are elements, however, of the existing design engineering software which can be evolved to function in the ULCE environment. The innovative aspects of the ULCE architecture require advances in software that may not be feasible until basic research questions have been answered. These questions are included throughout this section. Recent advances in artificial intelligence research (notably knowledge representation, natural language processing, expert system development, machine learning, and cognitive science) and product modeling provide the theoretical foundations for the solutions to these questions and for the development of the ULCE software.

The goal of the software described in this section is to implement the ULCE architecture by using a symbolic computing and object-oriented approach. The heart of the ULCE concept is the *Design-in-Progress*, a consistent representation of the engineering design data which is maintained throughout the life cycle of a product. By maintaining a computer-based comprehensive description of product design data, it is possible to customize the design process to meet the specific needs of the project (the *Meta-Design* concept). The Meta-Design concept involves designing the design process. The Meta-Design concept, teamed with the Design-in-Progress, distinguishes this architectural approach from traditional ones. The software requirements focus on these elements of the ULCE architecture.

The software environment will consist of mechanisms for *implementing instantiations of the ULCE design process* (via a sophisticated operating environment, an adaptive user interface, and standards for data exchange and software); *implementing the Design-in-Progress* (via methods for developing an extensible data representation for design engineering information and an advanced data base and knowledge base management system); and *implementing the Meta-Design concept* (via a multi-level optimization technique). Before describing these software components, a brief characterization of current CAE/CAD software is provided.

2. Current Computer-Aided Engineering Software

Despite noteworthy productivity improvements for specific design functions, current Computer-Aided Engineering (CAE) technologies have failed to significantly impact overall design effectiveness. Today's CAE tools were originally developed as aids to specific design or analysis functions and, consequently, are severely limited by:

- the inability to address a broad set of design requirements in a cost-effective manner,
- the lack of advanced information management capabilities for engineering design data and design process data,
- limited integration and communication between existing systems, and
- the lack of sophisticated application development capabilities.

The broad and interdisciplinary set of design requirements which must be considered or met during the design process has increased significantly in concurrence with the complexity of systems under design. Other factors also have contributed to the increase of design requirements. For instance, the DoD continues to place increased emphasis on reducing weapons system life cycle costs through improved reliability and maintainability. In fact, MIL-STD-499A (Engineering Management) requires that engineering specialties such as maintainability, reliability, and production engineering be totally integrated with design engineering. Although the number of requirements that must be addressed during the design process has increased significantly, current computer-based design tools lack the capability to address a large number of interdisciplinary design requirements in a cost-effective manner.

Current CAE/CAD tools were originally developed to perform specific design or analysis functions, not interdisciplinary functions. Examples of these tools include NASTRAN^{tm} for stress analysis, QUADPAN^{tm} for aerodynamic characteristics prediction, and Network Repair Level Analysis (NRLA) models for supportability characteristics prediction. These tools are usually developed using high-level programming languages, such as FORTRAN, and typically require hundreds of man-years to develop, maintain, and validate. This discipline-oriented approach to development has led to isolated systems with very limited communication capabilities. These isolated systems provide productivity improvements in a specific area, but they are not integrated to significantly improve overall design effectiveness. The results from one tool must either be manually re-entered into another system or transferred using a specialized translator that must constantly be updated.

The quantity of analytical tools for the development phases of a design is not considered a problem. Cataloging existing tools would produce an encyclopedic list. These analytical tools were typically automated using various versions of FORTRAN. To reduce the large data sets associated with these tools, engineers either had to write their own programs or to use programs written by computer science specialists or systems analysis groups. Several of these algorithms have achieved national acceptance and sponsorship by DoD interest groups.

For example, one of the current USAF preferred models is Network Repair Level Analysis (NRLA). NRLA was written by AFLC/LSS in FORTRAN, has been adapted for microcomputers, and works on a large data set with a unique file structure. NRLA, which is very effective in today's environment, will be used to illustrate difficulties in moving current software into the ULCE environment.

Assumptions concerning level of repair are routinely made by design engineers. The C-130 landing gear, redesigned for improved supportability as discussed earlier, considered repair technician skill level, support equipment, and the maintenance tasks desired to be performed by organizational and intermediate level people. These are all factors considered in NRLA.

The use of NRLA requires expertise beyond that of design. Few design engineers are trained, experienced, or have the time to deal with models such as NRLA. The following factors complicate the use of NRLA.

- The number of variables (in excess of 150) used in NRLA and their complex interactions require a degree of familiarity with the program's technical implementation details.
- The specialist (Supportability Engineer) that understands the model must build data arrays containing detail that is seldom available until after a significant number of design decisions have been made. Engineering judgment is used to predict parameters. Later analyses validate early level or repair decisions. The specialists use models like NRLA as a tool to help plan and investigate the impact of design decisions that may be 5 years in the future.
- NRLA requires specialists for its use and maintenance.
- NRLA does not use rules or principles that can be generalized.
- NRLA would require extensive redesign to operate in an on-line decision support environment.
- It is written in a classic procedural language that is driven by syntax.

- None of the current CAD software builds the unique data sets needed by NRLA.

Typically, current CAE and CAD systems provide limited mechanisms for integrating tools to form a cohesive environment. Thus customized translators must be built to share data among heterogeneous software. Customized translators are not cost-effective mechanisms for sharing data. Not only does this require excessive resources dedicated to translating data, it also requires an effective software development environment which can be tightly coupled to the automated design engineering environment. Most CAD/CAE tools provide interfaces to their working data bases. For example, CADAM^{tm} and CATIA^{tm} generate and maintain descriptions of designs that consist of geometric and associated data incorporated in either 2D/3D wireframe or 3D solid models. These models are usually maintained in local files or an internal data base, and the interface to these data is provided through high-level programming languages such as FORTRAN and C. Such interfaces are not sufficient to achieve tight coupling between CAD/CAE environments and software development environments.

Many assumptions are built into current CAD/CAE tools. For instance, most tools are developed to perform a specific function of a sequential design process. The tools in this case will not perform effectively if there is any deviation in the underlying sequential design process. Another set of assumptions typically built into a tool is knowledge of the design data, design rules, and design intent. Attempts to share this design data result in loss of the meaning and quantity of information from tool to tool. Often data must be manually reentered or regenerated. Methods for manually entering design data are notoriously tedious. No attempts are made to share information about design intent.

Many facets of the underlying architecture of current systems must be well understood by the user to function effectively with the software. The user interface often necessitates an understanding of the underlying program-specific data model and idiosyncratic access and command languages. Many systems require that the user understand what specific data is required, what software programs generate the information, where the code and data files are located in the file system, and other such programming details.

Although there are many weaknesses in existing CAD/CAE software tools which limit their effectiveness in an integrated design engineering environment, much work is underway to overcome these limitations and develop advanced concepts. Throughout the remainder of this section, special features of current design software which point toward

the future and may be evolved to function in the ULCE environment will be explored in the context of the software requirements for ULCE.

3. Implementing Instantiations of the Design Process

The ULCE software environment will be an integrated and layered set of software modules which facilitate the ULCE design process. *The ULCE system requires the development of a large system of interrelated software and data management capabilities.* To reduce the risk and cost of developing such a system, significant increases in software development productivity are needed. An object-oriented environment, specifically designed to support the engineering design process, will facilitate both the development and use of the software.

The ULCE system should integrate the functions that are currently provided by the host computer operating system, higher order languages, data base management, and user interface services. This system must be capable of running on future distributed, heterogeneous hardware systems, and should make this difficult environment invisible to the users. Some of the features that this environment should provide include:

- File access, protection, network integration ,and transparency.
- Methods for capturing design data. A number of methods should be developed to sketch a design, specify requirements, select a mathematical function from a library, etc. And, each of these methods will employ a layered approach to hide the details of implementation from the designer. For instance, one approach is to supply an application-specific higher-order language to the designer. A designer's sketch would be expressed using a task-oriented language. The high-level representation is compiled by the system. The compiled version is expressed in terms of lower-level primitives invisible to the designer. These primitives are stored in a rule base. This rule base would be hosted in a symbolic, non-procedural, object-oriented environment. The next lower level primitive would be in a language such as C, FORTRAN, assembler, or machine code.
- Analysis and simulation capability.
- Design integration, management, and control through a network representation of the design objects and attributes.

Current software characterized as rule based (ICAD^{tm}), sketch driven (Intergraph), handbook (Cognition), and CATIA^{tm} represent functionality that can be used to guide

ULCE software requirements. These representation or implementation choices are now mutually exclusive requirements for a new system.

To illustrate the current difficulty in merging automated tools which use diverse representation paradigms, the manner in which anthropometric analysis is performed by human engineers is described. Combiman (available on CADAM^{tm}) uses a wire frame human analogy to analyze the access and physiological characteristics of proposed designs. Current design graphic software often uses three-dimensional representations. It would be desirable to

(1) create a seamless interface between the wire-frame representations of Combiman and three-dimensional graphics packages, and

(2) to instill Combiman with intelligence to manipulate solid surfaces. (The program should have the capability to recognize that the operator is asking it to reach through a solid surface.)

ULCE could develop the link capability and use graphics to story board the results for reviewers and associated users of the design (training, etc.). Unfortunately, current technology does not allow direct linking of the Combiman capability to design software for either manual or imbedded analysis. Several problems must be resolved. First, it must be possible to directly process the graphic output of the programs without the need for human interpretation. Second, the complexity of the programs must be minimized so that operation or interpretation does not require specialists.

Specific components envisioned for the ULCE software environment include (1) the operating environment, (2) the user interface, (3) standard data exchange and software interfaces, (4) a conceptual model of engineering information and product data, and (5) an advanced knowledge and data base management system. The first three of these components are described below. The remaining components will be described in the context of the Design-in-Progress.

a. The Operating Environment

The operating environment is the layer of software that sits directly on top of the operating system (such as UNIX^{tm}). It interfaces directly to the operating system utilities (such as the file management system and the device drivers) and the other software modules of the ULCE software environment (such as the user interface, the knowledge-base and data base management system, etc.). Current CAE operating environments (also known as system shells, executive controllers, etc.) are similarly described.

The primary functions of the ULCE operating environment will be to control process invocation intelligently, coordinate design management activities, provide links to external tools, and hide implementation and lower-level details of the system.

To intelligently manage the invocation of distinct processes (such as the execution of an analysis program), the operating environment will use knowledge of the design process that was instantiated (via access to the knowledge-base that stores the design task objects); the configuration of the software modules as independent units and as a whole; and knowledge of both the state and contents of the Design-in-Progress. With this collection of knowledge, the operating environment will coordinate the operation of paralleled tasks and will facilitate the constant move among design process phases and levels of abstraction.

To coordinate design management activities, the operating environment will use the collection of knowledge listed above to monitor events, provide progress reports, provide version support, provide long-term archival capabilities, etc.

The operating environment will provide the hooks necessary to fully integrate (not merely attach) external software programs and data bases into the system. This will be achieved by utilizing knowledge of standard interfaces, by a capability to acquire and process knowledge concerning the external software (such as the information required and generated by the program), by a capability to access design data and feed the data to the external software, and by the capabilities of a built-in software-development environment. The success of the integration will be measured by the extent a user would need to be concerned with the idiosyncratic details of the tool. The most difficult aspect of developing this set of capabilities will be developing a method to configure the external tool within the ULCE environment without requiring excessive program development (such as translators) or restricting the external tool in any way.

Another important function of the operating environment will be to hide the complexity of the underlying operating system and other implementation details from the user without restricting the user from the use or functionality of system utilities. Details such as file organization, software location, operating system commands, etc., require knowledge that is not directly applicable to the task at hand (design). This transparency can be achieved only if methods are developed for assessing the cognitive complexity of operating environments with respect to particular tasks.

b. The User Interface

(1) User Interface Paradigms

Two features of the ULCE design process drive the need for significantly improved user interfaces to the system software:

- the quantity, content, and complexity of the data managed by the system for human communication, and
- the need to generate a large volume of object-oriented design relationship software.

The ULCE architecture calls for the capture and communication of much more computer-based data in the areas of requirements, functions, and decisions than are currently available. This data will be characterized by very complex interrelationships and high context. In order not to swamp design engineers in a data dump, the ULCE user interface to this data must be quite sophisticated. In today's design environment, at least two user interface paradigms (for data such as requirements) are being employed--a forms-driven interface to relationally stored data and a windowed icon-driven interface to data organized in object structures. The forms-driven interface is activated by a set of forms control keys that can be quite extensive. Typically, each form has an associated set of commands to augment the control keys. Users must master the use of both. Window-based interfaces were originally limited to personal computer-based operating systems and general purpose programs. Their use has migrated quite effectively to workstation-based CAE systems for graphics applications such as schematic-capture of logical designs but not necessarily for the capture and display of parametric data.

Several emerging software technologies should be pursued for application to the ULCE user interface. Among these are natural language interfaces to design data bases, context-oriented text search methodologies, and icon-driven user interfaces.

Natural language interfaces have proven useful in well-defined and limited domains. Their use in engineering environments needs to be investigated. If successful, the natural language interfaces will provide the flexibility needed to build complex access paths to design a data base at run-time. This is desirable because the forms style user interface largely restricts data access to paths established programmatically, thus limiting flexibility.

Some of the information in the ULCE process does not fit cleanly into the object-oriented or relational model (example; Lessons Learned text, bulk handbook data, and

complex numerical data). For these data types, the ULCE process will require sophisticated *context based access methods*, extensions of the methods used in the IBM STAIRS^{tm} (Storage and Information Retrieval System) program which dates from the 1960's.

An underlying object-oriented data structure greatly facilitates the implementation of *icon-driven user interfaces*. The applicability of icon-driven user interfaces to the display and capture of parametric data stored in object structures requires further investigation.

(2) Knowledge Capture

Creation of the ULCE design process will involve the capture of an explicit representation of virtually all design parameters and their relationships. *How to accomplish this information capture is one of the key implementation issues of the ULCE architecture.* While the underlying representation will be in an object-oriented language, it is not practical to train sufficient numbers of design experts in programming techniques to encode information about design parameters and their relationships. Similarly, we have seen many cases in the past where it is costly and time consuming to have discipline experts communicate program requirements to a programming team.

There are many ways in which knowledge capture may be achieved. One interesting possibility is to develop very high-level user interfaces to the underlying, object-oriented code-generation process. One or more user interfaces, tailored to a given discipline, might be necessary for each design discipline.

This approach can be illustrated by examples from current design systems. The ICAD^{tm} object-oriented design language (ICAD, Inc.) allows a class of designs to be specified as a set of design rules. This rule-based design can be quite powerful, for example, it is possible to derive rules from many design disciplines. After a particular design is created (instantiated) by specifying input parameters, the design can be enhanced to include a complete geometrical description, complete manufacturing information, and complete logistics information if the rule base is sufficiently complete.

While much of the ICAD^{tm} system is a forerunner of the future design process, the user interface must be vastly improved to be a model for the ULCE process. The user interface of ICAD^{tm} requires the use of a proprietary object-oriented language and Lisp. An example of what is possible for user interfaces is better illustrated by the Intergraph CAD system interface. Intergraph is also an object-oriented system, but geometrical

relationships are entered using an interactive graphical interface. As the design is drawn on the screen, the design intent is captured in object-oriented code, but the user has no sense of programming. Another example is the engineer's sketchbook program developed by Cognition. In this system, a designer can sketch a design and relate design features to terms in equations taken from standard engineering handbooks. Here too, the design is captured as object-oriented code, but the user interface buffers the engineer from the code generation process, and he sees only a graphical and perhaps intuitive interface.

These current examples are presented to illustrate that, while the ULCE process may be supported by extensive object-oriented code development, it will be possible to provide intuitive user interfaces which do not require expertise in object-oriented coding or other programming disciplines. The designers should not be required to be experts in object-oriented technologies. Research will be required to develop the necessary high-level interfaces for a wide range of engineering disciplines. Realistically, the final ULCE system will involve a complete range of interfaces, including some direct code generation (for those users who are responsible for low-level implementation details). Cognitive models of the users must be incorporated into the design of these interfaces to create a correspondence between the categories of users and the user interfaces of the various functions. It is also possible to leverage from on going research in machine-learning to develop adaptive user interfaces which, over a given period of time, will learn users' styles and adapt accordingly.

(3) Hardware Requirements

The software issue of powerful user interfaces drives at least one hardware requirement. ULCE software will involve high-level user interfaces which require significant computer processing power. This software requirement will, therefore, emphasize the move toward high-powered workstations as the primary end user equipment in the ULCE hardware architecture. Additionally, future design/engineering workstations will incorporate more ergonomic features such as screen height and slant adjustments, large color-graphic screens, and user-definable windows. Concurrent access to computer-based designs and advanced electronic mail/dialogue capabilities will allow design reviews between different locations/companies.

(4) Summary

The required user interfaces should be quite intuitive in their use and should be consistent throughout the ULCE design process. They should provide or use simple command languages, models of the system users, the capability to learn designers' styles, graphics for design information capture and data base manipulation, and knowledge about specific applications. This can only be achieved if the user interfaces are flexible.

c. Standards for Data Exchange and Software

(1) Data Exchange Standards

The key to a truly integrated design process is to streamline the current inefficient and paper-intensive process for exchanging product information. One goal in this respect is to minimize the amount of time required and resources used to translate data among functions in the design process. This will require the development of a computer-based product model (the foundation for the Design-in-Progress) to replace the engineering drawing as the primary mechanism for exchanging product information.

The mechanisms for exchanging data include hardware standards, communication protocols, and data exchange standards. In several cases, these capabilities in the ULCE environment will be built on top of existing and evolving capabilities. For example, the ULCE architecture will not require new hardware standards to replace IEEE 488 or EIA-232C for parallel and serial data communications, respectively. Additionally, the protocol standards, MAP/TOP and TCP/IP will be appropriate for ULCE. Ongoing work in the area of public domain product data standards (PDES, the Product Data Exchange Specification) and the corollary data exchange standards (IGES, the Initial Graphics Exchange Specification, and EDIF, the Electronic Design Interchange Format) will be affected by ULCE requirements.

The ongoing work in the area of product definition and data exchange standards is currently in a state of flux. There are several independent efforts focusing on the development of product data standards for specific applications (such as mechanical engineering, integrated circuit design, and printed circuit board design). The work of these independent efforts is beginning to overlap in specific areas. It is too soon to determine how the overlaps are going to be resolved. The PDES effort, still in the stage of research and development, is aiming to develop a complete definition of product data which is independent of particular applications. If successful, PDES will encompass all of the

existing exchange standards (such as IGES and EDIF). To achieve a complete definition of product data, there are many research issues to be addressed. Among these issues is the question of the extent to which the conceptual-product model must incorporate object-oriented concepts. These open issues are described further in Subsection F.4.a.

The need to exchange product data with less sophisticated vendors, team members, or customers will require new standards that define application-specific views of the product model (such as schematic views, layout views, manufacturing views, etc.). These views will define the interfaces across multidisciplinary application- and function-specific tools. An understanding of the symbolic transformations required to translate between views is required to develop these standards.

(2) Software Standards

For several reasons, software standards pertaining to object-oriented languages, analytical software, and operating systems need to be developed for ULCE software as opposed to generic standards such as ANSI FORTRAN. Such development must also follow a coherent strategy.

The primary reasons are interoperability and transportability. Interoperability is required between design and analytical tools and between the tools and a specified set of operating systems. The latter requirement is driven by a pragmatic assumption concerning various hardware vendors, hardware-specific operating systems, and the need for competitive procurement.

Transportability of analytical tools and design representations is seen as an increasingly important requirement in a ULCE environment. Centralized development of tools in DoD is expected to continue along with contractor developed tools that must be delivered and used by the government. A good example is the vulnerability and survivability models developed by Mitre. They can be directed contractually with much less cost as readily transportable software. Translating designs for specific numerical control equipment or for unique graphics requirements will require unique software, which increases cost.

The suggested strategy for building standards is to integrate their development with ULCE process development. A phased approach in which existing standards were identified for convergence with ULCE requirements as part of their evolution would be

complemented by incrementally building new standards specified by designers not programers. Incrementally developing standards would have several benefits, including:

- minimizing false starts for emerging software,
- avoiding the conventional sequential approach to standards development, and
- building consensus in both government and industry.

Ada was probably the first language to be built in a similar manner.

Incremental development would allow a top-down approach with increasing levels of detail as the requirements become better understood and as a consensus is built. This strategy also is well suited for the open architecture ULCE.

The emergence of object-oriented programming techniques that can be applied to design software is another reason for incremental standards. The ULCE design tool standard would influence language developers which, in turn, would affect an object-oriented language standard.

4. Implementing the Design-in-Progress

a. Developing an Extensible Representation of Design Engineering Data

Engineering design data is fundamentally different from traditional business data. Design data is characterized by a rich set of data types (matrices, parametric data, tables, text, graphical images, etc.), complex interactions between features of the design, partial design descriptions (due to the interactive nature of design), design data which may be correct during an early stage but incorrect at a later stage, multiple versions of one design, and previous states of the design. Furthermore, engineering design data is hierarchical (designs may be or may consist of assemblies) and may have multiple aspects (mixed representations such as behavioral and geometric layout aspects). Design knowledge includes constraints on design concepts (imposed by reference, domain, or design rules), goals (context-dependent constraints described at a specific level of abstraction), design experience, and design process knowledge.

As suggested by these characteristics, a representation (or *model*) of engineering design data must incorporate features of the network, hierarchical, relational, and object-oriented data model types. With the network model type, data entities (or attributes) may be linked together in a network structure to represent relationships, hence constraints. The

hierarchical model type, a special case of the network model, provides a structure for representing design hierarchy (the assembly of objects to create super-objects). The relational model type provides a structure for organizing tabular and parametric data. The object-oriented model type provides a structure which focuses on entire objects, their properties, the operations that can be performed on them, and their constraints. One way to combine the applicable features of these four model types is to define objects as either groups of relations, subsets or collections of networks, or the parent nodes of hierarchies. Constraints can be realized as relationships between attributes of design objects (thus incorporating features of the network model type).

The ULCE architecture will use object-oriented design tools that describe products in terms of design rules, constraints, and requirements. An underlying data model for engineering design data provides a foundation for the integration of these design tools and the knowledge and data base management system.

The model for engineering design data must have the expressive capabilities to support the characteristics of design engineering data. The key issue which must be addressed is the level at which the design data and engineering knowledge should be represented by the data model. In other words, can a sufficient set of primitive design objects be identified so that all other design data can be derived from those primitive objects? The primitive objects and the semantics (what the objects represent) must be unambiguously defined. Furthermore, all developers of the object-oriented design tools and data bases must agree to the same set of primitives and meanings. If other underlying data models are used for the design tools, then mappings between the nonstandard data models and the ULCE standard data model must be established.

Product models are representations of *specific types of products* such as a landing gear. Product models are specific instances of generalized (or conceptual) data models described above. The product model provides the structure in which the Design-in-Progress data is built. Valid designs are generated by adding design data (balanced design requirements) to the product model framework. Thus, the Design-in-Progress is an instantiation of a particular product model.

Methods to define and validate both conceptual data models and specific product models must be developed. These methods will include some type of language for describing the contents and structure of both the conceptual data model and the specific product models.

The capability to structure both conceptual engineering design data and specific product data will provide a platform for the implementation of the advanced data management and decision support capabilities of the ULCE architecture.

b. Implementing Advanced Knowledge-base and Data-base Management Systems

Engineering data must be shared among a multiplicity of application programs used in the design process. Currently, the most common way of sharing engineering data is to file-transfer data from one computer program's data files to another program's files. File transfers of point-in-time copies are done because most application programs used by engineers today are file-processing programs. In this mode of operation, each separate program defines and controls its own data. However, this approach has obvious disadvantages when several such programs need to make concurrent, interactive updates of shared files.

CAE and CAD systems are migrating toward data base-processed concepts. In this mode of operation, data base management system (DBMS) software is used for the centralized administration of shared engineering data as well as the concurrent manipulation of that data by many interactive users. DBMS software includes some means for physically structuring data on mass storage devices, transaction processing, access control, concurrency control, crash recovery, and, of course, data base manipulation.

The three traditional data base models (hierarchical, network, and relational) have been used in various combinations for engineering applications. Although relational DBMSs have slower response times than hierarchical or network DBMSs, the use of the relational model has gained popularity for scientific and engineering applications.

There are commonalities in the management of business-related data and engineering design data. These include:

- The data base management system must be flexible.
- The data base management system must support working data bases that are optimized for individual tools/users and global data bases for data that is shared by all tools and users.
- The data base management system must interface with existing software.

- The data base management system must represent all of the kinds of knowledge needed for the application (i.e., design of landing gear), manipulate objects to derive new objects, and incorporate new knowledge into the system.
- The data base management system must access external data.

There are significant differences, however, between the management of business-related data and engineering data. Some of these difference are:

- Engineering data must be organized across multiple representations of the same design. This requires managing successive versions of authorized data and managing successive versions of design alternatives, most of which may never be authorized for release. To implement these functions, temporal relationships must be captured and processed.
- Design requires the re-use of previous designs or design components.
- A data base management system for design data must understand the phased and iterative nature of the design process.
- A data base management system for design data must understand the multi-aspect nature of design data.
- A data base management system for design data must handle different levels of abstraction.
- A data base management system for engineering data must support the growth of library data and access library data from external sources.
- The data base management system must organize and control a large amount of parametric data (design requirements and relationships).
- The data base management system must distinguish between design data, data about the design process, library data, and system data.

An object-oriented data model, which uses appropriate features of the network, hierarchical and relational data model, and temporal relationships, will facilitate the management of engineering design data. Engineering applications are typically concerned with objects. That is, an engineer is usually concerned with the design and analysis of some structured entity or object, such as a collection of parts that comprise an assembly. Objects are manipulated as a logical group for the creation, access, manipulation, and storage of engineering data.

Design objects represent subsets of the design and auxiliary structures for grouping together object representations into useful clusters. Hence, objects can be nested within objects, forming a hierarchy of design data. For example, a complete airplane can have a

fuselage, wing, etc. A fuselage can have a forward section, mid section, etc. A designer can request access to the entire design or to any of its subparts.

Logically-associated engineering objects can be manipulated on diverse, physically separated computers and workstations. While logically perceived as a central data base of engineering data, the data will be physically distributed to optimize response times. A distributed approach, especially if implemented using parallel architecture (as done on Tandem's Non Stop SQL data base), provides a much more uniform interface for the designer.

Before an advanced, distributed data base management system, as described above, is developed, a thorough understanding of the data base management functions (or responsibilities) for engineering design data must be attained. The key function that must be well understood is the management of complexity. The data base will contain a large amount of complex data (many types of objects with unique behaviors and complex interrelationships and interactions) which must be accessed efficiently and non-deterministically. The data base management system must effectively filter the data so that only relevant portions of the design are accessed/processed. The standard interfaces described earlier will be incorporated to guide this filtering function.

To effectively manage complexity, the stored level of detail of the design primitives must be appropriate for the ULCE design process. Both the conceptual data model (used to represent design engineering data) and the product models (used to represent the design data for a particular product) will affect the level of detail at which data is stored in the data base.

There are many other advanced functions which should be incorporated in the data base management system. These include inferencing mechanisms to reason with and reach conclusions about the design data, mechanisms for building and maintaining historical knowledge or design experience, and automatic methods for verifying the contents of the data base and the rules used to reason about the data.

5. Implementing the Meta-Design Process

The Multilevel Optimization Using Linear Decomposition (MOLD) computer program, developed at Lockheed-Georgia, has demonstrated the technical feasibility of the proposed ULCE architecture design decision process planning approach (referred to as the Meta-Design process). MOLD, written in Common Lisp, uses heuristics based on a simple

model of the design decision-making process. With this model, the design process decision structure is determined by the extent the design features impact the set of driving requirements. To formulate the design decisions, the following heuristic rule is used:

> "interacting design features having equal impact on the driving requirements must be considered together, as part of an individual design decision."

This simplistic approach should be thoughtfully examined from the point of view of management science planning, scheduling, and decision theory.

A limitation of the approach represented in MOLD is that the formulation of design decisions is based on design optimization as a decision support tool. Development of enhanced computer-aided, problem-formulation techniques must go hand-in-hand with research on decision support for design.

6. Summary

The ULCE design environment will be an integrated and layered set of automated design software which will support all aspects of the ULCE architecture. This software will facilitate the management of complex design parameters and relationships, the planning of the design process, and the execution of the design process. The following components must be developed to achieve this environment.

- *An advanced operating environment.* The operating environment will provide intelligent control of process invocations (both serial and parallel), coordination of design management tasks, integration mechanisms, controlled access to all components of the ULCE environment, and system transparency. To perform these functions, the operating environment makes use of many types of knowledge--knowledge of the product under design (generic); the design process (as planned); design management functions; the state and contents of the Design-in-Progress representation of the design; the information requirements for each of the automated tools, system utilities, and system configuration information. By making use of this knowledge, the operating environment will facilitate constant movement among the design process phases and levels of abstraction. The following functions of the operating environment will require further research:

  (1) The operating environment must make use of sets of complex knowledge to manage the environment.

  (2) The operating environment must know what design management tasks are required and when they are appropriate.

(3) The operating environment must integrate external tools into the system without requiring massive software development efforts or limiting the capabilities of the tools.

(4) The operating environment must hide the complexity of the underlying operating system without limiting the user.

- *An adaptable user interface.* The user interface must provide flexible communication mechanisms to capture and display computer-based design data. The design data is characterized by very complex interrelationships and high context. Natural language interfaces, context-oriented text search methodologies, and icon-driven user interfaces are emerging software technologies which should be explored for application to the ULCE user interface. The following functions of the user interface require further research:

(1) Capturing and displaying complex and interrelated data.

(2) Providing application-specific languages, such as the U.S. Air Force VHSIC Hardware Design Language (VHDL) appropriately to capture design knowledge and intent.

(3) Providing a graphics-oriented language for accessing the data base.

(4) Incorporating cognitive models of the various users of the system to provide adaptable user interfaces.

- *Standards for data exchange and software.* Standards are necessary to streamline the inefficient and paper-intensive process for exchanging product data. Mechanisms for exchanging data include hardware standards (i.e., EIA-232C), protocol standards (i.e., MAP/TOP), and data exchange standards (i.e., IGES, EDIF). The ULCE environment will leverage from existing and emerging capabilities. In the area of application-independent product data representation, no comprehensive standard exists. The PDES, Product Data Exchange Specification, effort is aiming to develop a complete, application-independent definition of product data. ULCE requirements should impact the development of PDES. The following requirements for data exchange and software standards need further research:

(1) A standard computer-based representation of product data which replaces the engineering drawing.

(2) Mappings between established data exchange standards and a standard definition of the product data.

(3) Software standards for object-oriented languages, analytical software, and operating systems.

- *Conceptual and product models of engineering design data.* An extensible representation of design engineering data, which incorporates features of the network, hierarchical, relational, and object-oriented data model types, is needed for ULCE. This data model will provide a foundation for the integration of the ULCE design tools and the knowledge/data base management systems. To provide a standard definition of the design data needed throughout the life cycle of a product, the following areas need further research:

  (1) Appropriate data models for the representation of conceptual design engineering data and product-specific data.

  (2) Effective levels of abstraction for design data and engineering knowledge representation.

  (3) Methods to define and validate both conceptual and product data models.

- *An advanced data base management system.* The advanced data base management system will control the virtual centralized administration of shared engineering data and the concurrent manipulation of the data by many interactive users. It will organize the engineering data and design knowledge across multiple representations of the same design (design alternatives) and multi-disciplinary functions. Areas which require further research include:

  (1) The management of complexity in a run-time environment.

  (2) The identification and characterization of data base management tasks.

  (3) Methods for reasoning with and verifying the contents of the data bases.

  (4) Automatic methods for storing and retrieving historical design data (design experience and previous designs).

- *Mechanisms for implementing the Meta-Design process.* The ULCE architecture design decision process planning approach requires automated tools to (1) model the structure of the decision process and (2) utilize heuristics to determine the extent to which a design feature impacts the set of driving requirements. These tools must be fully integrated with the ULCE operating environment, the design representation (the Design-in-Progress), the data base management system, and the automated analysis tools. The manner in which the decision process structure is formulated and manipulated needs further exploration.

Recent advances in the application of symbolic computing, object-oriented, and product modeling technologies are being made as a result of both government and industry efforts. Two examples are the U.S. Air Force Engineering Information System (EIS) program and the Ontologic VBase product. The EIS program will use symbolic computing

technologies to prototype a framework for the integration of current and future design automation tools. This framework will consist of a set of services and reference specifications necessary to achieve an integrated design engineering environment. The services and specifications will address tool integration and portability, data interchange, engineering management and control, information management, and EIS administration. (*The Department of Defense Requirements for Engineering Information Systems*, November 25, 1987). Ontologic, a Massachusetts-based company, has developed a commercial object-oriented data base, VBase, for CAD/CAM applications. This database provides many advanced mechanisms necessary to model and manage complex engineering and product data. These and other related efforts will contribute significantly toward both the development of software described and the realization of the ULCE architecture.

G. AN EVALUATION OF THE PROPOSED ULCE ARCHITECTURE

1. Introduction

In this section, the ULCE architecture presented in the previous section will be evaluated, and its feasibility, likely improvements in the design quality, cost of implementation, and potential life-cycle cost savings will be discussed. Problems in the current design process will be illustrated and features of the proposed architecture that eliminate or alleviate these problems also will be discussed. Emphasis will be placed on identifying issues that will impact the implementation of the ULCE architecture. The hardware and software issues that have been identified are discussed in more detail in the following section.

Two of the three major sections of the ULCE architecture shown in Figure E-4 are felt to be evolutionary, requiring logical growth of existing or planned approaches. These two sections are the *Generate Design Alternatives* and *Make Design Decisions* procedures. Advanced computing technology can be applied to these sections with advantage, but they are still evolutionary. The Meta-Design approach embodied in the *Plan Design Decision Process* is the key revolutionary approach proposed. Central to this approach is the idea that a design process, in particular the decision-making process, should be driven by the peculiar requirements and system definition of a program. This approach does not have a model in today's design environment, and therefore involves the major implementation issues of the ULCE architecture. Particular emphasis is given to this phase.

In evaluating the ULCE architecture, the feasibility and cost of implementation are considered for each section. Potential improvements in design quality and potential life cycle cost savings of improved designs are a function of the overall design process and are discussed in the overall context of the proposed architecture.

2. General Philosophy of the ULCE Design Process

The need for an ULCE architecture has grown out of the recognition of the need to consider supportability and producibility in earlier design phases and at the same level as performance, cost, and schedule. The potential advantages for this approach are numerous and have been discussed in many places, therefore, need not be enumerated here. What is important here is that it be recognized that reaching this goal involves incorporating many more design considerations into every phase of the design process. Logically then, this

requires more design information to be managed, leading to increased coordination and communication problems. It can be easily argued that, in many cases, the current design process does not handle the current load of performance, cost, and schedule tradeoffs well.

The ULCE architecture problem addresses the bigger issue of reducing the limitations in the existing design process, thereby making it possible to consider more requirements during all design phases. Therefore, ULCE improves the design process for the full range of design considerations including, but not limited to, supportability and producibility. The proposed Meta-Design approach to engineering design does not explicitly show supportability and producibility processes, but it properly groups them with the full set of possible requirements. By allowing the design process to be modified based on a program's requirements, the Meta process fully supports incorporation of the desired mix of performance, supportability, producibility, cost, and schedule considerations.

a. Approach to Problems

The primary difference between the Meta architecture and current design practice is that design problems are addressed, and this change is the revolutionary part of the procedural architecture. In the current process, a design alternative can be given great depth of analysis by several disciplines before a conflict in the original requirements or system definition is found. Often design decisions are made unilaterally by one discipline with the assumption that they do not effect other disciplines. When conflicts arise they can take on the nature of a problem if a great deal of effort has been wasted or if much effort is needed to resolve the conflict. Individual or organizational ego can play a large part in the elevation of a design conflict to problem status.

In the proposed architecture, as implemented in the Plan Design Decision Process, an organized attempt is made to recognize and manage all interrelationships in the design process. The implications of this statement and the issues raised are discussed in detail in a later section. Having an explicit representation of the design parameters and relationships is the central issue of the Meta approach. The essence of the Meta approach is to use the explicit relationships to establish decision points and to evaluate sufficient design alternatives to support these decisions. In this approach, the design process is controlled in a breadth-first search of the design space. Various design alternatives are pursued only to the level necessary to support a previously identified decision and when backtracking is minimized.

Obviously some unforeseen design relationships will be discovered. It is believed that these, on the average, will have less negative impact on the design as the more significant relationships will become visible in the process. It will be important in the proposed approach to capture unforeseen relationships and to incorporate them into the explicit knowledge base to improve the next design. In this way the system will become "smarter" as time passes and more experience is gained.

b. Capture of Design Intent

An evolutionary, but key, feature of the ULCE architecture is the emphasis on the capture and communication of design intent. The current design process centers around the system definition and other forms of design information are viewed as tools used in localized areas and localized design phases. This results in problems with communicating the design across each of the major interfaces--conceptual to preliminary, preliminary to detailed, engineering to manufacturing, and to the customer. In the proposed architecture, requirements, functional decomposition, system description, analysis results, and decisions provide the primary communication paths in the design process. It is intended that dissemination of these data be made as widely as possible within the limits of security concerns. In fact, communicating this information without swamping design engineers in a data dump is one of the key issues of the proposed architecture.

3. Technical Evaluation

In the following sections, each section of the ULCE architecture will be evaluated for feasibility and cost to implement, and key design issues will be identified.

a. Generate Design Alternatives

Figure E-7 illustrates the procedural flow within the Generate Design Alternatives section of the ULCE architecture. In today's design environment, each of these functions is done with varying degrees of formality and computer support. Because the changes proposed in this area are largely in implementing details, the Generate Design Alternatives procedural section of the ULCE architecture is considered to be an evolutionary growth of the current design process.

The ULCE architecture requires significant increases in both the volume and content of the data managed and in the level of integration within the generation of design alternatives. Both the functional analysis and the system definition procedures are input

drivers of the Plan Design Decision Process and must be well integrated to maximize the utility of the ULCE process. The required integration of this, and downstream ULCE procedures, would tax the current approach which includes relational data bases and support codes written in a traditional procedural language like FORTRAN.

The Generate Design Alternatives procedure is evolutionary because the process is done today, but its implementation in the ULCE architecture calls for development of a programming environment that is a significantly ahead of today's capability. In the progression of programming languages, there has been a continual move toward syntax that more closely models the problems to be solved. This raises an issue that we feel is one of the key underpinnings of the proposed ULCE architecture:

> *the need for a completely integrated programming environment in which the object-oriented language is highly coupled with what is traditionally considered the operating system as well as an integrated database management system and provisions for very high level user interface support.*

This programming environment must be capable of operating in a heterogeneous hardware environment and must make the hardware transparent to the programming user. While the Meta Architecture approach is conceptually valid, *we believe that implementation would not be successful without a significantly improved software environment.*

The entire process described as Generate Design Alternatives must have strong human involvement to provide the creative portion of the design process. What distinguishes the ULCE process from today's environment is the projected increases in data volume, the content that must be captured and communicated, and the integration of the system definition procedure with the Design Decision Planning Process. Because of the magnitude of the task, the ULCE software must identify all design parameters and relationships around which the design decision architecture will be built. The correct set of relationships must be related to the functional decomposition, and a system description must be developed for each design alternative. Developing this highly interrelated system with today's procedural languages would be a complex task. The process seems to be much more manageable in an integrated object-oriented environment.

The current lack of standards for object-oriented languages must be addressed very early in the ULCE process development. It will be difficult to commit to a major system development effort without strong standards on the underlying software system. Standardization efforts have already begun but it may not be adequate to wait for the fruits

of this effort. A strong case could be made for developing an object-oriented programming environment directly related to the design process. A more detailed study of the advantages and disadvantages of this approach needs to be made. The underlying software environment has a strong impact on the entire ULCE architecture and decisions on its development need to be addressed early.

(1) Implementation Cost for Design Alternative Generation

There is no simple way to estimate the cost of implementing the necessary object-oriented software environment or the Generate Design Alternatives section of the ULCE architecture on which it is built. One approach to this estimation problem would be to survey a number of computer-language development efforts and to relate them to the estimated scope of the integrated object-oriented system envisioned. A particularly interesting number would be the cost of developing the Ada language in recent years. The ULCE development is likely of the same magnitude if the complete Ada environment is considered.

As mentioned, many of the software components in the Generate Design Alternatives section of the architecture exist in some organizations and are under development in others. The combined Requirements Flowdown and Functional Analysis procedures are currently under development at Lockheed Aeronautical Systems and it is estimated that they will take approximately 6 to 8 man-years to implement using current relational data base and procedural language. If implemented in an integrated object-oriented software environment, this estimate might be reduced to 3 to 4 man-years.

(2) Evaluation of the Plan Design Decision Process

The heart of the Meta-Design concept of the proposed ULCE process is found in the Plan Design Decision Process. This is the section of the ULCE process that custom designs a design execution and decision process based on the requirements, functions, and system description of each particular design problem. This section is considered revolutionary because there is no direct parallel in today's design process. In fact, in today's process, the design decision planning process typically evolves very slowly over a period of time. For this reason, the process cannot adapt to subtle changes in the design requirements, so relationships are overlooked and problems occur.

Successful development of the Meta-Design is the central issue of the ULCE architecture. The major underlying issue or enabling technology is the capture of explicit

representation of all design relationships that might affect a given class of design problems. Because these relationships must be formulated in computerized form, this task is formidable, however, the object-oriented approach appears to make this task feasible. The MOLD research tool has already used symbolic processing to demonstrate the ability to handle a wide range of types of design relationships ranging from simple numeric relationships to those represented by large-scale computer codes. Conceptually, these relationships can be extended to handle any representation, including those only manageable as human interactions. Both quantitative and qualitative parameters and relationships must be managed by the Design Decision Planning Process.

As the design parameters and relationships are generated, it will be important to integrate them with the functional decomposition and system description processes. The task of identifying the correct set of relationships for each phase of the design process will not be cost effective unless it is supported by an automated process. Here again, an object-oriented programming environment offers to provide the required complex interconnection with the chosen design alternatives in a cost effect manner. A simple example will illustrate this point. As design parameters and relationships associated with aircraft wings are identified, they are coded as class and instance variables and methods of a "wing" object. When the system description is expanded to include a wing, all of the required parameters and relationships are automatically associated with the design alternative. When the Design Decision Process is planned (decomposed), all required information is easily identified and brought into the planning process.

A prototype for the design process planning has been built as a research tool, known as MOLD (Multilevel Optimization using Linear Decomposition). While this tool has validated the concept of the Meta-Design approach, there are still several issues that require additional development, including

- Developing decision strategies,
- Establishing how sensitivities are passed between tasks, and
- Scheduling design tasks.

Development of design decision strategies is a major enabling technology of the proposed ULCE architecture. The system must provide a range of decision support tools, including the capability to handle both qualitative and quantitative decisions. Much more research is needed in the area of making qualitative decisions, particularly in the area of assessing sensitivities of qualitative decisions, to changes in design parameters.

At the current stage of its development, MOLD includes only relatively simple quantitative parameters. Methods are being developed to handle the required passing of sensitivities between design tasks. This work has identified the problems associated with sensitivities of complex quantitative parameters and with qualitative parameters. There is considerable research underway in the area, including the work by Sobieski at NASA Langley, which address techniques for calculation of sensitivities for complex quantitative parameters.

While methods for scheduling design tasks is a development issue, it is not considered to be a problem. The scheduling rules currently in the MOLD program have identified some shortcomings in the simplistic approaches attempted so far. The final solution will probably involve a combination of several approaches, including knowledge-based methods.

The cost of capturing all of the required explicit design relationships is difficult to estimate. Experience with simple design problems using the MOLD program has indicated that approximately 2 man-hours per design relationship is reasonable for simple numeric (textbook level) examples. Before a reasonable estimate of the overall task can be made, more experience will be required with a wider range of complex relationships.

c. Make Design Decisions

The Make Design Decision Process section of the ULCE architecture is considered to be evolutionary because the basic process exists in some form in all design organizations. In fact, the required process could be done with no computer support or at least no more than existing numerical optimization routines. This entire section of the proposed ULCE architecture could be implemented as a human process although a loss of system response would occur.

The major implementation issue identified in this section is the need for improved decision-support tools. The overlap with the Decision Planning Process is natural because the detailed choice of the decision support tool to be used in one design task is made in that process while the tool is actually applied in this process of the ULCE architecture. Of particular importance is the ability to integrate qualitative and quantitative decision support tools in one design task. The desired level of integration has not been demonstrated in a design environment and must be developed. It must be remembered that parts or all of this process can be implemented as human procedures and that there is no real limit on successful implementation of this section of the architecture.

4. Likely Improvements in Design Quality

Improvements in the quality of designs produced by the proposed ULCE architecture fall into two categories:

- A general move toward a more optimum mix of the important design parameters.
- Earlier recognition and solution of problems in the design.

Engineering experience provides a strong conviction that both of these categories will yield significant improvements, but they are quite difficult to quantify for some future program. In fact a product, in many cases, is forced to operate in an environment that is significantly different from the requirements to which the product was designed.

The ULCE concept has grown out of the recognition of the need to incorporate supportability and producibility consideration from the beginning of the design process. The concept has expanded to recognize that the current design process has not always handled the mix of requirements from traditional performance, cost, and schedule well. Simple inclusion of additional requirements might yield improvements in the areas that receive increased emphasis, but, more than likely, this approach would create additional problems in other areas.

Creation of the optimum mix of capabilities can only occur through engineering creativity, but the architecture of the ULCE process must assure an orderly consideration of all design relationships. All pertinent relationships must remain visible for human consideration, and none must "fall through the cracks." The proposed architecture supports this need in two ways. The Generation of Design Alternatives and Decision Making processes are both designed to capture and communicate far more information (design intent) than is done in today's environment. By making more of this information available to a wider audience of involved design engineers, a more detailed evaluation of all alternatives is achieved. Hence, quality of designs (here defined as the best mix of capabilities) will improve.

a. Example

The view of quality as elimination of problems is easier to illustrate in detail by use of past bad examples where the design process failed and problems arose. These problems can range from those that caused redesign in the normal design process cycle to those that were not discovered until the product was in service. Everyone knows of many of these

horror stories, and every design team has been involved in more than one example. As a natural result of individual and organizational ego, a great deal of valuable design knowledge is lost when these problems go undocumented or downplayed.

The potential for the ULCE process to reduce or eliminate problems can be illustrated (and several issues identified) in an example taken from an installation of an antenna on the empennage of the C-130 aircraft. A top-level requirement for the system stated that the antenna be jettisonable. This requirement grew out of a very early concern that, in an antenna structural failure, the controllability of the aircraft might be jeopardized. The system design, therefore, included a set of pyrotechnic cable cutters that could be activated to release the antenna. After the antenna system design was complete, it was decided to perform a safety analysis of the system. The need for the safety analysis was largely driven by the existence of the pyrotechnic devices. The analysis concluded that the presence of the pyrotechnic cutters introduced far more safety and maintenance concerns than the original antenna cable failure. There was no traceability to the originator or to the justification of the original requirement and, at the point of the analysis, it was more costly to modify the design than to leave the cutters in place.

How would the proposed ULCE architecture have eliminated this problem, and what issues are raised? In the ULCE architecture, the formal documentation and communication of requirements and functional decomposition receive strong attention. In ULCE, the jettison requirement would have been justified and linked to a higher level requirement. Better visibility alone might have identified the potential problem much earlier than it was actually caught. One issue that is raised by this scenario is that the requirements traceability should extend all the way back into the original customer requirements setting process. There is a somewhat artificial, though necessary, disconnect imposed by a contract. Future work on the ULCE approach should address means of making the entire requirements flowdown and review process as seamless as possible. In some circumstances, this effort will have security implications, and a global view of the requirements hierarchy may have to be severely restricted.

The ULCE Meta-Design process would potentially have eliminated the antenna jettison problem in a second way. When the system description had progressed to the point of identifying pyrotechnic devices, the cutter "object" would inherently have associated safety analysis relationships. The Design Decision Planning would have highlighted the need for this consideration much earlier than it actually occurred. Had this problem been identified earlier, a decision process would have begun before the design was finalized, and

other solutions would not have been locked out simply due to time and cost constraints. One issue raised here is *the need for tremendous detail in the generation of design relationships for all objects in the class of designs being considered.*

5. Potential Life Cycle Cost Savings

It is difficult, if not impossible to realistically predict the potential life cycle cost savings afforded by implementation of the ULCE process since this savings will be a function of the set of requirements placed on each unique design problem. The life cycle cost savings are likely to accrue from engineered reductions in manufacturing costs and improved (lowered) support costs, as well as simple elimination of problems in both the engineering and manufacturing phases. It could be argued that many of these savings could be attained in today's environment simply by increasing the emphasis on producibility and supportability. Conceptually, additional requirements could be placed on the design process and life cycle savings would result.

The real impact of the ULCE architecture, in particular the requirements and decision planning procedures, will be to allow the additional desired requirements to be added to the process without creating additional problems and with less of an adverse impact on other design considerations (performance, cost, and schedule) than is currently possible.

In order for the later phases to begin with a design definition that has fewer inconsistencies that could lead to later problems, the potential impact of extending the early design phases (conceptual and preliminary) should be considered as part of the process of developing the ULCE architecture. Possibly everyone in engineering is aware of the data that show that a very large percentage of the committed design is based on a very small percentage of the total design effort. While the ULCE architecture addresses maximizing the effectiveness of the relatively small front-end effort, the possible advantages of simply extending these important design phases should be considered.

Several approaches to studying the potential of extending the early design phases are possible. The most straight forward is a study of past major design efforts. In this study, problems that arose in later design phases would be identified, and the cost and time to resolve the problem would be noted. It would be necessary to identify those problems whose solutions would have been different if they were discovered earlier and to estimate the cost and time benefits of these preferred solutions. The savings associated with the problems that could have been discovered in earlier design phases would then be weighed

against the cost of longer time spans in early phases. It is conceivable that the total time for the design and manufacturing phases would not be increased, only reallocated.

This review of real design efforts would only be practical if an egoless view of the design history is possible. Bias due to both individual and organizational ego would have to be considered. It also may be possible to adequately model or simulate a typical design process for a class of design problems instead of reviewing actual case histories.

H. CONCLUSIONS AND RECOMMENDATIONS

The objective of this study has been to provide a top level view of Unified Life Cycle Engineering as it might be applied to a specific design problem--Landing Gear Design. It is hoped that this approach has resulted in an architecture which reflects real world considerations and is responsive to the many problems which must be addressed by the design community when designing a complicated system, such as a landing gear. While much of the material developed in this effort is specific to landing gear design, the ULCE architecture which has been developed appears to be fairly general in applicability. In the following sections, conclusions and recommendations are given which relate to limitations of the current design process and desired architectural features for a ULCE system (in particular, addressing communications among designers and engineering specialists, design decision making, and design data base integration.) Finally, recommendations for future research directions and implementation strategies are given.

1. Conclusions

a. Limitations of the Current Design Process

The current design process for landing gear (and for many other defense systems) is limited in the following ways:

- *It is driven by scheduled deliverable data items.*

  Pressure to turn out drawings leads to a design process which is optimized for a depth-first search of the design space leading to a rapid build-up of design detail. Such a search provides an early lockout of many design alternatives which may be better from the standpoint of producibility and/or supportability.

- *Its engineering is labor-intensive.*

  Much of the required design detail is generated manually. Even computer aided tools such as CADAM^{tm} require direct manipulation by highly skilled engineers whose time could be better spent in attacking difficult conceptual or technical problems.

- *The sequence of design decisions is inflexible.*

  The order in which decisions are made is driven largely by the cost of producing design detail rather than a prioritization of design requirements. Decisions requiring a higher level of definition are pushed to the end of the process.

- *Producibility and supportability enter relatively late in the process.*

  This is because current methods require a high-level of design definition before producibility and supportability can be considered. Unfortunately, as the design progresses, residual design freedom rapidly decreases. By the time producibility and supportability are considered, there is very little that can be done to change the design without a very expensive and time-consuming redesign effort.

- *Design data is fragmented.*

  There are multiple representations of a design concept, such as solid models, finite element meshes, and standard drawings, which are supplied to subcontractors or the manufacturing department. Consistency of these representations is difficult to maintain.

- *As the design transitions from phase to phase, information is often lost.*

  Design intent is not always clearly inferred from drawings, and the connection between design features and the original requirements is not always transparent.

- *Communications among design team members and engineering specialists is inadequate.*

  Considerations of producibility and supportability currently enter the design process via the requirements generation and design review processes. Direct communication of design concepts by engineering specialists to the design team is limited.

b. Desired Architectural Features of an ULCE System

An effective ULCE design process must alleviate most of the problems noted above. The following are features of such a system which appear to be needed to solve such problems.

- *A team approach to generation of conceptual design alternatives is needed.*

  Engineering specialists in the various *ilities* must be allowed to contribute their ideas regarding design features early in the design process. While these ideas may not be feasible with respect to all design requirements, by bringing such ideas to the attention of designers, the chances are greatly improved that some feasible variation in the design which incorporates such features can be developed and that design could be superior from the standpoint of producibility, supportability, or another of the *ilities*.

- *Computerized support is necessary to generate design alternatives.*

  If more people are allowed into the design game, as suggested above, management of design alternatives will become more complicated. Thus, a computerized system (probably object-oriented in nature) will be necessary to enable documentation, rapid retrieval, and manipulation of design concepts.

- *Explicit Design Process Planning must be based on an analysis and understanding of the design problem to be solved.*

  The design process must not be static or taken as given. It must be driven by design requirements, design methods, and available technology. Automated tools are needed which will allow explicit generation of a sequence of design tasks and their durations and resource requirements which, if executed, will result in a high probability of success in the synthesis of a design meeting the required goals and objectives of the development. These tools must take into account explicitly the structure of the system to be designed (as identified in the alternatives generation process) and the interrelationships among design parameters and characteristics. Moreover, this planning system must allow for efficient midstream reconfiguration of the design process when customer requirements change or when advancing technology presents new opportunities to the design team.

- *The design decision process must have orderly and controlled execution.*

  In order to carry out the design process plan in an efficient manner, tools for efficient creation of design detail (parametric synthesis) and effective management of design data (configuration management and control) must be in place. Documentation of design decisions and their rationales must be generated and made available to all persons who will be affected by these decisions (especially downstream functions such as manufacturing and support.)

- *A unified design information base (called the Design-in Progress in the preceding chapters) is needed which provides an integrated description of the design at each stage of the process and which is accessible by all parties involved in generation or analysis of the design.*

  This information base would contain (but not necessarily be limited to) data on requirements, functional descriptions, physical hierarchy, geometry data, tolerances, materials data, parts data, manufacturing process plans, logistics support analysis data, and design history data. Such an information base would help to solve the problems of fragmentation in design data in the current process and would be essential for effective communication among the design team and other specialists as the design progresses.

2. Recommendations

Based on the conclusions drawn from the study, the study team makes the following recommendations regarding ULCE research and development, and implementation strategies.

RESEARCH AND DEVELOPMENT ISSUES

- *Additional research on human interactions in design should be conducted.*

  A key element of the ULCE architecture is a team approach to development of design alternatives. Questions arise on how best to implement such a team concept. Which *ilities* engineers should be on the team, and how many? What kind of techniques could be employed to help such a team reach consensus in the face of the large number of design decisions and factors which must be considered. What are the lessons to be learned from previous team design efforts relating to approaches which are successful versus those that are not?

- *Theory, methodology, and tools need to be developed for design process planning, execution, and control.*

  Significant research has been directed at manufacturing process planning, but little at design process planning. One of the problems in planning the design process is that in the manufacturing field many of the activities are well defined and easily structured; however, design is an open ended process by nature and is characterized by creativity and invention. It is becoming clear that proper sequencing and execution of design decisions is a key factor in obtaining designs which are properly balanced from downstream as well as up front considerations. Key disciplines which would be involved with this research would include general design theory and methodology, operations research, decision science, management science, and artificial intelligence. Because it is the feeling of the study team that design process planning must be driven by actual design requirements, methods, and physical reality, it is clear that to be successful, advances in this field must be coupled closely with specific domain knowledge in the various engineering disciplines. Thus, to be successful, work in this area must be multidisciplinary in character.

- *Research should be conducted in the areas of data base management systems, data modeling, and applications of object-oriented technologies to design systems.*

  These fields are key to development of the Design-in-Progress information base which is the foundation for the ULCE architecture. Research needs to be directed not only to application of object-oriented techniques with representation of physical design data (parts, subassemblies, and assemblies,

etc.) but also to representation of design requirements and functional hierarchies needed in systems engineering, and for representation of design tasks in design process planning, analysis, and control.

- *User interface issues are critical in making ULCE a viable environment for design of producible and supportable systems.*

  While object-oriented and symbolic computing technology may be required as the foundation for a mature ULCE system, the designer must be shielded from such technical details and allowed to work in a way that is natural for him. If a particular designer is comfortable in a LISP environment, he should be allowed to work in it. However, many designers will not be comfortable in such environments. The job of a mechanical engineer, for example, should be focused on design and analysis of mechanical systems, not on computer programming. Research is needed on development of user interfaces which will allow designers to be as creative and productive as possible with a minimum requirement for learning computer-specific languages or protocols.

- *Techniques for automated generation of design detail must be further developed.*

  Currently, there are several computer-aided engineering systems which provide for parametric syntheses of design geometry. Under the ULCE architecture advanced in the previous chapters, there will be a continual requirement for changing designs, creating new alternatives, and creating variants of existing alternatives. Without automated tools for quickly developing adequate design detail for analysis of these alternatives, the ULCE design process will be prohibitively expensive in terms of time and manpower resources. In IC design, silicon compilers have already provided a great deal of automation in generating design detail. Similar capabilities must be developed in other engineering disciplines.

IMPLEMENTATION ISSUES

- *There is a critical need for development of a comprehensive, phased plan for development and application of ULCE related technologies and implementation of ULCE supportive practices.*

  This plan must address what is feasible and reasonable for the government to do, what must be accomplished within industry, and what the government can do to stimulate industry to do their part.

- *People issues, such as implementation of team concepts in design of defense systems, can be addressed now and can show significant payoffs with minimal requirements for technology development.*

Implementation of a team approach to design requires first and foremost a strong top management commitment to do what needs to be done to make the approach successful. Secondly, it requires breaking down barriers of communication between the various specialty communities and the design community within each company (and within the government).

- *The design community must play a key leadership role in ULCE development and implementation.*

While a great emphasis now is being placed on injecting requirements for producibility and supportability into the design process, it must be recognized that the job of the designer is very difficult, even just considering questions of performance, cost, and schedule. Development of additional requirements to be laid on the designers by *ilities* specialists operating in relative isolation will not solve the problem. A cooperative effort, lead by the designers themselves, must be undertaken.

REFERENCES

1. "Implementation Plan for Unified Life Cycle Engineering," February 1987.

2. "Special Report: On Good Design," *IEEE Spectrum*, pp. 27-72, May 1987.

3. "Report of the Joint Industry-DoD Task Force on Computer Aided Logistic Support (CALS)," IDA Report R-285.

4. Norman S. Currey, Aircraft Landing Gear Design--Principles and Practices, AIAA Education Series, to appear.

5. "Unified Supportability Design Requirements," Lockheed-Georgia Project Memorandum LG87PM7249-001, May 15, 1987.

6. T. Smithers, "Alvey Design to Product Demonstrator," a paper presented at the Third International Conference on Advanced Information Technology: Artificial Intelligence in Manufacturing--Key to Integration?, held in Switzerland on 7-8 November 1985, Edinburgh University

7. Popplestone, R., T. Smithers, J. Corney, A. Koutsou, K. Millington, G. Sahar, "Engineering Design Support Systems," a paper presented at the First International Conference on Applications of Artificial Intelligence to Engineering Problems, Southhampton, 15-18 April 1986, Edinburgh University.

DISTRIBUTION
IDA PAPER P-2063

ARCHITECTURE AND INTEGRATION REQUIREMENTS FOR AN ULCE DESIGN

189 Copies

	Number of Copies
Department of Defense	
Department of Defense Office of the Under Secretary of Defense (Acquisition) Rm. 3E933, Pentagon Washington, DC 20301	
ATTN: Hon. Robert B. Costello	1
Dr. Robert C. Duncan Director of Defense Research and Engineering Room 3E1006, Pentagon Washington, DC 20301	1
Office of the Under Secretary of Defense for Acquisition Directorate, Program Integration Room 3E1034, Pentagon Washington, DC 20301-3000	
ATTN: Thomas P. Christie	1
Deputy Under Secretary of Defense, Research and Advance Technology Room 3E114, Pentagon Washington, DC 20301	
ATTN: George P. Millburn	1
OUSD (R&AT/ET) Rm. 3D1089, Pentagon Washington, DC 20301-3080	
ATTN: Raymond Siewert	1
OUSD (R&AT)/ET Rm. 3D1089, Pentagon Washington, DC 20301-3080	
ATTN: Dr. Leo Young	1

OUSD (R&AT/CET)
Rm. 3D139, Pentagon
Washington, DC 20301

ATTN: E. D. Maynard, Jr., Director 1

Office of the Deputy Under Secretary of Defense for
Tactical Warfare Programs
Rm. 3E1044, Pentagon
Washington, DC 20301-3100

ATTN: Donald M. Fredericksen 1

Defense Science Board, OUSD(A)
Room 3D1020, Pentagon
Washington, DC 20301-3140

ATTN: Eugene G. Fubini 1

Defense Systems Management College
Office of the Commandant
Ft. Belvoir, VA 22060-5426

ATTN: Brig. Gen. Charles P. Cabell 1

Office of the Under Secretary of Defense (Acquisition)
Deputy Under Secretary of Defense, Strategic and
Theater Nuclear Forces
Rm. 3E130, Pentagon
Washington, DC 20301-3090

ATTN: Lawrence Woodruff 1

Department of Defense
Assistant Secretary of Defense, Production
and Logistics
Rm. 3E808, Pentagon
Washington, DC 20301-8000

ATTN: Jack Katzen 1

DASD (Procurement)/OASD (P&L)
Rm. 3E144, Pentagon
Washington, DC 20301-8000

ATTN: Eleanor R. Spector 1

Mr. Russell R. Shorey 1
Assistant Deputy, Systems, OASD/P&L
Department of Defense
Room 2B322, Pentagon
Washington, DC 20301-8000

DASD, Production Support
Rm. 3E144, Pentagon
Washington, DC 20301-8000

ATTN: Robert C. McCormack 1

OASD (P&L) PS/IR
Director, Industrial Resources
Rm. 3C257, Pentagon
Washington, DC 20301-8000

ATTN: Richard E. Donnelly 1

OASD (P&L) PS (IPQ)
Director, Industrial Productivity and Quality
Rm. 2A318, Pentagon
Washington, DC 20301

ATTN: Jack Strickland 1

Mr. John A. Mittino 1
DASD (Logistics)
Rm. 3E788, Pentagon
Washington, DC 20301-8000

Office of the Secretary of Defense (P&L)
Director, CALS
Rm. 2B322, Pentagon
Washington, DC 20301

ATTN: Michael McGrath 1

OASD (P&L), Systems
Director, Weapon Support Improvement Group
Rm. 2B322, Pentagon
Washington, DC 20301

ATTN: Martin Meth 1

Col. Larry Griffin 1
OASD (P&L) WSIG
Rm. 2B322, Pentagon
Washington, DC 20301-8000

Mr. Joseph Arcieri 1
Deputy Director, DoD Weapon Support
Improvement and Analysis Office
1400 Two Skyline Place
5203 Leesburg Pike
Falls Church, VA 22041

Dr. Raymond S. Colladay 1
Director, Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209-2308

Dr. George L. Donohue 1
DARPA/ISTO
1400 Wilson Boulevard
Arlington, VA 22209-2308

Dr. William E. Isler 1
Director, Prototyping Applications
DARPA, ISTO
Arlington, VA 22209-2308

Dr. Phillip A. Parrish 1
DARPA/DSO
1400 Wilson Boulevard
Arlington, VA 22209-2308

Dr. Richard A. Reynolds 1
DARPA/DSO
1400 Wilson Boulevard
Arlington, VA 22209-2308

Dr. Jacob Schwartz 1
DARPA/ISTO
1400 Wilson Boulevard
Arlington, VA 22209-2308

Dr. William Bandy 1
DARPA/ISTO
1400 Wilson Boulevard
Arlington, VA 22209-2308

SDIO/D
Rm. 1E1081, Pentagon
Washington, DC 20301-7100

ATTN: Lt. Gen. James Abrahamson 1

SDIO/T
Rm. 1E148, Pentagon
Washington, DC 20301-7100

ATTN: Col. Garry Schnelzer 1

SDIO/S
Rm. 1E149, Pentagon
Washington, DC 20301-7100

ATTN: Maj. Gen. Eugene Fox 1

Office of the Secretary of Defense 1
OUSDRE (DoD-IDA Management Office)
1801 N. Beauregard Street
Alexandria, VA 22311

Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2

Miscellaneous, U.S. Government

Mr. Dave Bettwy National Bureau of Standards Rm. B108, Bldg. 233 Gaithersburg, MD 20899	1
Dr. Robert Hocken National Bureau of Standards Automated Production Technology Division Meteorology Bldg., Rm. B106 Gaithersburg, MD 20899	1
Mr. Ted Lettes, Director Small Business Technology, Liaison Division Office of Productivity, Technology and Innovation U.S. Department of Commerce Room 4827, Commerce Building Washington, DC 20230	1
Dr. John Singer Director, AMRF National Bureau of Standards Gaithersburg, MD 20899	1
Dr. Jaroslaw Sobieski Deputy Head, Interdisciplinary Research Office NASA Langley Research Center Structures Directorate Hampton, VA 23665	1
Dr. John Zuk Chief, Advanced Plans and Programs Office NASA Ames Research Center Moffett Field, CA 94035	1
Mr. Brad Smith National Bureau of Standards Gaithersburg, MD 20899	1
Dr. Tony Woo Program Director, Computer-Integrated Engineering National Science Foundation Design, Manufacturing and CIE 1800 G Street, N.W. Washington, DC 20550	1

Dr. Michael J. Wozny 1
Director, Division of Design, Manufacturing,
and Computer-Integrated Engineering
National Science Foundation
1800 G Street, N.W.
Washington, DC 20550

Department of the Army

Honorable John W. Shannon 1
Assistant Secretary of the Army,
Installations and Logistics
Rm. 2E614, Pentagon
Washington, DC 20310-0103

Mr. Eric A. Orsini 1
Deputy Assistant Secretary of the Army
Office of the Assistant Secretary of the Army (I&L)
Rm. 3E620, Pentagon
Washington, DC 20310

Honorable Jay R. Sculley 1
Assistant Secretary of the Army, Research,
Development and Acquisition
Rm. 2E672, Pentagon
Washington, DC 20310-0103

Gen. Carl Buono 1
Chief of Staff, Army
Rm. 3E668, Pentagon
Washington, DC 20310

Deputy Chief of Staff, Research, Development
and Acquisition
SARD-ZS
Washington, DC 20310-0103

ATTN: Maj. Gen. Cianciolo 1

Dr. Kenneth J. Oscar 1
Assistant Deputy CS, Development,
Engineering and Acquisition
HQ, Army Materiel Command
5001 Eisenhower Avenue
Alexandria, VA 22333-0001

Mr. Edwin Greiner 1
Assistant Deputy
Army Materiel Command
Rm. 10S06
5000 Eisenhower Avenue
Alexandria, VA 22337

Mr. Clarence Meese 1
Belvoir Research and Development Center
Code STRBE-TQR
Ft. Belvoir, VA 22060-5606

Mr. Geza Papp 1
Chief of Technology
U.S. Army AMCCOM
Building 62
Picatinny Arsenal
Dover, NJ 07806-5000

Department of the Navy

Honorable Everett Pyatt 1
Assistant Secretary of the Navy, Shipbuilding
and Logistics
Rm. 266, Crystal Plaza #5
2211 Jefferson-Davis Highway
Arlington, VA 22202

Mr. W. J. Willoughby 1
Director for Reliability, Maintainability
and Quality
Office of the Assistant Secretary of the Navy
(Shipbuilding and Logistics)
Washington, DC 20360-5000

Honorable Thomas F. Faught, Jr. 1
Department of the Navy
Assistant Secretary of the Navy, Research,
Engineering and Systems
Rm. 4E732, Pentagon
Washington, DC 20350-1000

Vice Admiral Stanley R. Arthur 1
Deputy Chief of Naval Operations for
Logistics
Department of the Navy
Washington, DC 20350-2000

Mr. Emerson Cale 1
Deputy Director
Chief of Naval Operations
Rm. 4B546, Pentagon
Washington, DC 20250

Department of the Air Force

Hon. John J. Welch 1
Assistant Secretary of the Air Force (Acquisition)
HQ, USAF, The Pentagon
Washington, DC 20330-1000

Deputy Assistant Secretary, Acquisition,
Management and Policy
The Pentagon
Washington, DC 20330-1000

ATTN: Mr. Daniel Rak 1

Lt. Gen. C. C. McDonald 1
HQ, USAF/LE
Washington, DC 20330-5130

Brig. Gen. F. S. Goodell 1
Special Assistant for Reliability and Maintainability
HQ, USAF/LE-RD
Rm. 4E259, Pentagon
Washington, DC 20330

Assistant Deputy Assistant Secretary of
the Air Force (Acquisition)
Washington, DC 20330-1000

ATTN: Maj. Gen. Donald L. Lamberson 1

Mr. Oscar Goldfarb 1
Deputy for Supply and Maintenance
SAFAL - Rm. 4D865, Pentagon
Washington, DC 20250

Dr. John Dimmock 1
Technical Director
Air Force Office of Scientific Research
Bolling AFB, Bldg. 410
Washington, DC 20332

Dr. Sam Rankin 1
Director, Mathematical Optimization Program
Air Force Office of Scientific Research
Bolling AFB
Washington, DC 20332-9448

Capt. Maureen Harrington 1
Program Manager, ULCE Decision Support
Logistics and Human Factors Branch
Air Force Human Resources Laboratory
Wright-Patterson AFB, OH 45433-5000

Capt. Donald Lowdermilk 1
PM RAMCAD Program
Air Force Human Resources Laboratory, LRA
Wright-Patterson AFB, OH 45433

Col. Donald Tetmeyer 1
Director, Logistics and Human Factors Division
Air Force Human Resources Laboratory
Wright-Patterson AFB, OH 45433-5000

Lt. Col. Joseph Coleman 1
Air Force Human Resources Laboratory/LRA
Area B, Bldg. 190
Wright-Patterson AFB, OH 45433

Dr. Harris Burte 1
Chief Scientist
AFWAL, Materials Laboratory
Wright-Patterson AFB, OH 45433

Mr. Alan E. Herner 1
Operations Research Analyst
Air Force Human Resources Laboratory
Logistics and Human Factors Division
Building 190
Wright-Patterson AFB, OH 45433-5000

Mr. Raymond L. Johnson 1
Technical Director
Aeronautical Systems Division
USAF, ASD/ENS
Directorate of Systems Engineering
Wright-Patterson AFB, OH 45433-6503

Dr. Melvin C. Ohmer 1
Field OPR, ULCE Program
AFWAL Materials Laboratory
AFWAL/CAM
Wright-Patterson AFB, OH 45433

Mr. Frederick T. Rall 1
Technical Director, ASD/EN
Wright-Patterson AFB, OH 45433

Mr. Mike Silverman 1
AFWAL/ILTO
Wright-Patterson AFB, OH 45433

Mr. Nick Bernstein 1
AFWAL/FIBR
Bldg. 45
Wright-Patterson AFB, OH 45433

Dr. Alan Burkhard | 1
AFWAL/FIEE
Wright-Patterson AFB, OH 45433

Dr. John Hines | 1
AFWAL/AAD-UPO
Wright-Patterson AFB, OH 45433

Capt. John Thomas | 1
AFOSR/MM
Wright-Patterson AFB, OH 45433

Mr. Anthony Coppola | 1
RADC/RBET
Griffiss Air Force Base
Rome, NY 13441-5700

Lt. Terry Oxford | 1
RADC/RBET
Griffiss Air Force Base
Rome, NY 13441-5700

Mr. James R. Meeker | 1
Air Force Systems Command/DLSR
Bldg. 1535, Rm. CD310
Andrews Air Force Base
Washington, DC 20334

Col. Eugene Tatini | 1
Air Force Systems Command/PLX
Andrews Air Force Base
Washington, DC 20009

Industrial Organizations

Dr. Shapour Azarm | 1
Department of Mechanical Engineering
University of Maryland
College Park, MD 20742

Mr. John M. Barker | 1
Director of Product Support
Boeing Aerospace Company
P.O. Box 3999, MS 82-09
Seattle, WA 98124-2499

Mr. George Beiser | 1
Private Consultant
3001 N. Florida Street
Arlington, VA 22207

Dr. Ben Blanchard 1
Assistant Dean of Engineering for Extension
Virginia Polytechnic and State University
College of Engineering
341 Norris Hall
Blacksburg, VA 24061

Mr. Jim Brimson 1
Vice President, Business Development
CAM International, Inc.
611 Ryan Plaza Drive, Suite 1107
Arlington, TX 76011-8098

Dr. Dale E. Calkins 1
Research Associate Professor
Ocean Engineering Program
Department of Mechanical Engineering
University of Washington
Seattle, WA 98195

Ms. Kathryn M. Chalfan 1
Artificial Intelligence Specialist
Boeing Computer Services
Artificial Intelligence Center - ATAD
P. O. Box 24346
Seattle, WA 98124-0346

Mr. Howard Chambers 1
Vice President for Logistics
Rockwell International
100 N. Sepulveda Boulevard
El Segundo, CA 90245

Dr. Lynn Conway 1
University of Michigan
College of Engineering
Chrysler Center, Room 263
Ann Arbor, MI 48109-2092

Mr. Darrell Cox 1
Rockwell International
201 North Douglas Street
El Segundo, CA 90245

Mr. Michael Davis 1
General Dynamics, Convair Division
P.O. Box 85357, MZ P1-2610
San Diego, CA 92138

Dr. John R. Dixon 1
Program Director, Design Theory and Methodology
National Science Foundation
Design, Manufacturing, and CIE
1800 G Street, N.W.
Washington, DC 20550

Mr. E. Perley Eaton 1
Principal Engineer, Federal Systems
TRW Defense Systems
Command Support Division
1 Federal Systems Park
12900 Fair Lakes Parkway
Fairfax, VA 22033

Mr. Mark Erdrich 1
Laboratory Head
Adv. CAD/CAM/CAE
Eastman Kodak Company
7/23 Kodak Park
Rochester, NY 14650

Mr. Ralph Ezard 1
Vice President, Consulting Operations
Automation Technology Products
1671 Dell Avenue
Campbell, CA 95008

Dr. Walter Fabrycky 1
Head, IE/OR Department
Virginia Polytechnic and State University
Department of Industrial Engineering and
Operations Research
Blacksburg, VA 24061

Dr. Stephen Fenves 1
Carnegie-Mellon University
Civil Engineering Department
Schenley Park
Pittsburg, PA 15213

Dr. Iman Foroutan 1
Chief, Engineering Automation Section
Microelectronic Circuits Division
Industrial Electronics Group
Hughes Aircraft Company
P. O. Box H, 500 Superior Avenue
Newport Beach, CA 92658-8903

Dr. Robert E. Fulton 1
Professor, Department of ME
Georgia Institute of Technology
Mechanical Engineering Department
Atlanta, GA 30332

Mr. Ned Glassman 1
Assistant Manager, Adv. Programs
Support Laboratory
Hughes Support Systems
P. O. Box 9399, Bldg. A1
Long Beach, CA 90810-0399

Mr. John C. Goclowski 1
Director, Advanced Systems
Dynamics Research Corporation
60 Frontage Road
Andover, MA 01810

Mr. Siegfried Goldstein 1
Siegfried Enterprises, Inc.
P. O. Box 2308
North Babylon, NY 11703

Mr. Eric Hausner 1
Logistics Engineer
TRW Defense Systems Group
Systems Engineering and Development Division
119B/4034, One Space Park
Redondo Beach, CA 90278

Mr. Jim Hayes 1
Group Engineer, LSA
Lockheed California Company
P.O. Box 551
Burbank, CA 91520-7278

Mr. Jerry Hollingsworth 1
Boeing Aerospace Company
P.O. Box 3999, MS 82-09
Seattle, WA 98124-2499

Mr. Ken Johnson 1
Manager, Design Technology Department
Lockheed-Georgia Company
Department 72-92/Zone 419
86 S. Cobb Street
Marietta, GA 30063

Mr. Benjamin Kaminsky 1
President, CAM-I
611 Ryan Plaza Drive, Suite 1107
Arlington, TX 76011-8098

Dr. Clinton Kelly 1
Corporate Vice President for Advanced
Technology Programs
SAIC, Inc.
Rm. 1408
1710 Goodridge Drive
McLean, VA 22102

Mr. Thomas L. Kelly 1
Assistant Division Manager
Quality-Products Operations Division
Hughes Aircraft Company
Radar Systems Group
P. O. Box 92426
Los Angeles, CA 90009

Dr. Haim Kennett 1
Vice President, Research and Engineering
Boeing Aerospace Company
7705 Marginal Way S., M/S 82-47
Seattle, WA 98124

Dr. Mohammad A. Ketabchi 1
Assistant Professor, EE and CS
Santa Clara University
Santa Clara, CA 95053

Mr. C. T. Kitzmiller 1
Advanced Technology Center
Boeing Computer Services
P.O. Box 24346, MS 7L-64
Seattle, WA 98124-0346

Mrs. Mary Klement 1
Engineering Specialist
General Dynamics
P.O. Box 85357, MS P1-2610
San Diego, CA 92138

Mr. Albert Knight 1
Naval Ocean Systems Center
Computer Integrated Engineering, Code 936
San Diego, CA 92152-5000

Dr. Janusz Kowalik 1
Engineering Technology Applications
Boeing Computer Services
2760 160th Avenue, S.E.
Belleview, WA 98008

Dr. Robert Kuenne 1
Professor of Economics
Princeton University
63 Bainbridge Street
Princeton, NJ 08540

Mr. W. D. Lewis 1
General Dynamics Corporation
12101 Woodcrest Executive Drive
St. Louis, MO 63141

Dr. Philip E. London 1
Vice President, Expert Systems Development
and Founder
Cognition, Inc.
900 Tech Park Drive
Billerica, MA 01821

Mr. Seaforth M. Lyle 1
Vice President
Ontologic, Inc.
47 Manning Road
Billerica, MA 01821

Mr. Lee R. Madison 1
Staff Engineer
Advanced Supportability Engineering Department
Lockheed-Georgia Company
Marietta, GA 30063

Mr. Stephen A. Magnus 1
Editorial Assistant, CAD/CAM Alert
Management Roundtable, Inc.
824 Boylston Street
Chestnut Hill, MA 02167

Mr. Walter W. Maguire 1
Staff Vicc President, Quality Management
GM/Hughes Aircraft Company
Corporate Offices
7200 Hughes Terrace, P.O. Box 45066
Los Angeles, CA 90045-0066

Mr. John Manzo 1
Corporate Manager
Systems Development Process
Digital Equipment Corporation
Maynard, MA 01754-2571

Mr. Peter Marks 1
Vice President, Product Planning
ATP, Inc.
1671 Dell Avenue
Campbell, CA 95008

Dr. Barry McNeill 1
Arizona State University
College of Engineering and Applied Sciences
Department of Mechanical and Aerospace Engineering
Room ERC 577
Tempe, AZ 85287

Dr. Michel A. Melkanoff 1
Director, Manufacturing Engineering Program
University of California at Los Angeles
Boelter Hall 3532
Los Angeles, CA 90024-1600

Mr. Richard C. Messinger 1
Vice President, Chief Technical Officer
Cincinnati Millacron
Cincinnati, OH 45209

Mr. Frederick J. Michel 1
Private Consultant
8409 Felton Lane
Alexandria, VA 22308

Dr. Alan Mitchell 1
Director, Preliminary Design Tool Development
Research and Engineering Division
Boeing Aerospace Company
P. O. Box 3999, MS 82-23
Seattle, WA 98124-5214

Dr. Ronald S. Morris 1
Director, Product Assurance (ATF Project)
Northrop Corporation
Aircraft Division
#1 Northrop Avenue
Hawthorne, CA 90250

Dr. Gerald Nadler 1
Chairman, Department of Industrial and Systems Engineering
University of Southern California
Industrial and Systems Engineering
University Park
Los Angeles, CA 99089-1452

Mr. Joseph Naft 1
Director, Computer Aided Design Laboratory
Engineering Research Center
University of Maryland
College Park, MD 20742

Mr. James L. Nevins 1
Division Leader
The Charles Stark Draper Laboratory, Inc.
Robotics and Assembly Systems Division
555 Technology Square
Cambridge, MA 02139

Mr. Guy B. Olney 1
Boeing Computer Services
P.O. Box 24346, M/S 7L-62
Seattle, WA 98124

Mr. David Owen 1
NTL, Inc.
P. O. Box 597
Northport, NY 11768

Dr. Michael Pecht 1
Associate Professor of Mechanical Engineering
Mechanical Engineering Department
University of Maryland
College Park, MD 20742

Mr. Joseph Piteo 1
United Technologies
Sikorsky Aircraft
MS S308A
6900 Main Street
Stratford, CT 06601-1381

Dr. Robin J. Popplestone 1
Director, Laboratory for Perceptual Robotics
University of Massachusetts
Department of Computer and Information Science
Lederly Graduate Research Center
Amherst, MA 01003

Dr. Alan L. Porter 1
Professor, Georgia Tech
School of Industrial and Systems Engineering
Atlanta, GA 30332

Mr. Arne P. Rasmussen 1
Private Consultant
464 Severnside Drive
Severna Park, MD 21146

Mr. Edward Rogan 1
Design Technology Department
Lockheed-Georgia Company
Department D72-92
Marietta, GA 30063

Dr. Lawrence Rosenfeld 1
President, ICAD, Inc.
1000 Massachusetts Avenue
Cambridge, MA 02138

Mr. Charles R. Runyan 1
Assistant Manager, Engineering Division
Hughes Aircraft Company
Radar Systems Group
P. O. Box 92426, Bldg. R1
Los Angeles, CA 90009

Brother Tom Sawyer 1
Bishop McNamara High School
6800 Marlboro Pike
Forestville, MD 20747-3270

Dr. Daniel Schrage 1
Professor, School of Aerospace Engineering
Georgia Institute of Technology
School of Aerospace Engineering
Atlanta, GA 30332

Mr. Stanley M. Stuhlbarg 1
Manager, Advanced Programs
Hughes Aircraft, Industrial Electronics Group
Microelectronic Circuits Division
P. O. Box H, 500 Superior Avenue
Newport Beach, CA 92658-8903

Mr. Jay M. Tenenbaum 1
Schlumberger Fellow
Schlumberger Palo Alto Research
3340 Hillview Avenue
Palo Alto, CA 94304

Dr. Delbert Tesar 1
Professor of Manufacturing, Robotics, and Logistics
University of Texas
Department of Mechanical Engineering
Room ETC 4.146C
Austin, TX 78712

Ms. Carol Tierney 1
General Dynamics Land Systems
P.O. Box 1804, MS 498-0503
Warren, MI 48090

Dr. Edison T.S. Tse 1
Director, Decision Systems Laboratory
Stanford University
Department of Engineering-Economic Systems
Stanford, CT 94305

Dr. John N. Warfield 1
Director, Institute for Advanced Study in the Integrative
George Mason University
4400 University Drive, 219 Thompson Hall
Fairfax, VA 22030-4444

Dr. John M. Wellborn 1
Manager, Maintainability and Human Factors
General Electric Company
1000 Western Avenue
Lynn, MA 01910

Mr. Arthur M. Westerberg 1
Director, Engineering Design Research Center
Carnegie-Mellon University
Pittsburg, PA 15213-3890

Mr. Robert I. Widder 1
Private Consultant
3203 Thornapple Street
Chevy Chase, MD 20815

Mr. David F. Zarnow 1
Assistant Manager, Hybrid Engineering Dept.
Hughes Aircraft Co., Solid State Products Div.
Microelectronic Circuits Division, MS A2202
500 Superior Avenue, Bldg. 700
Newport Beach, CA 92658-8903

Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Gen. William Y. Smith	1
Mr. Philip L. Major	1
Dr. William J. Schultis	1
Dr. Victor A. Utgoff	1
Dr. Jeffrey H. Grotte	1
Dr. Frederick R. Riddell	1
Mr. William E. Cralley	10
Ms. ML Brei	1
Dr. Karen J. Richter	1
Mr. David A. Dierolf	1
Mr. G. Watts Hill	1
Control and Distribution	10

END
DATED
FILM
8-88
DTIC